UF | Industrial & Systems Engineering
UNIVERSITY of FLORIDA

Generating scalable training data for learning to cut problems via set packing and covering problems

Zixuan Feng, Aleksandr Kazachkov, *University of Florida*

MIPLIB is not enough for *ML for MIP* problems

- **Goal:** Train models that are helpful for **generic** MIP solving. *(This is different from models that perform well on structured data.)*
- *MIPLIB* is expert-selected library meant to give researchers access to the breadth of problems solved in practice.
  - **Assumption:** *MIPLIB* is a sample from a distribution of real-world instances.
  - **Problem:** *MIPLIB* Collection only contains 1065 instances with 217 remaining open, while ML models need a lot of data to train.

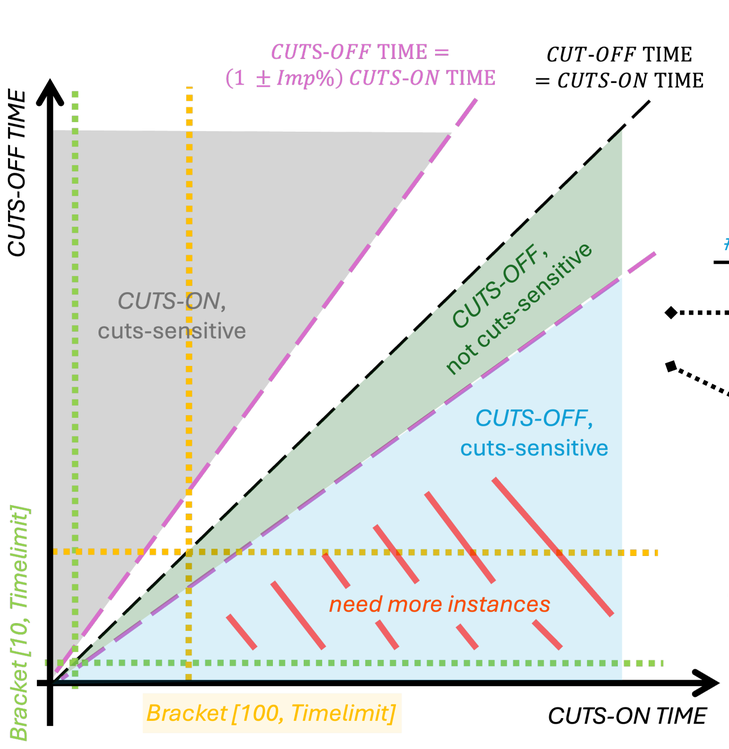
We do not know how to draw from the *MIPLIB* distribution.

Is there a set of scalable generators from the same distribution as *MIPLIB*?

- Many scalable generators aim for this task: *Ecole*[1], *G2MILP*[2], *MILPBench*[3], …
- *Distributional MIPLIB*[4] (*DMIPLIB*) aggregates generators from 13 domains, classified into different hardness levels.

Case study: Does learning about cuts from DMIPLIB generalize?

- **Task:** Predict whether an instance will benefit from disabling cuts (*CUTS-ON* vs *CUTS-OFF*).
- **Experimental setup**: Instances from *MIPLIB 2017 Collection* and *DMIPLIB*; solve with *PySCIPOpt* with a time limit of *3h*.



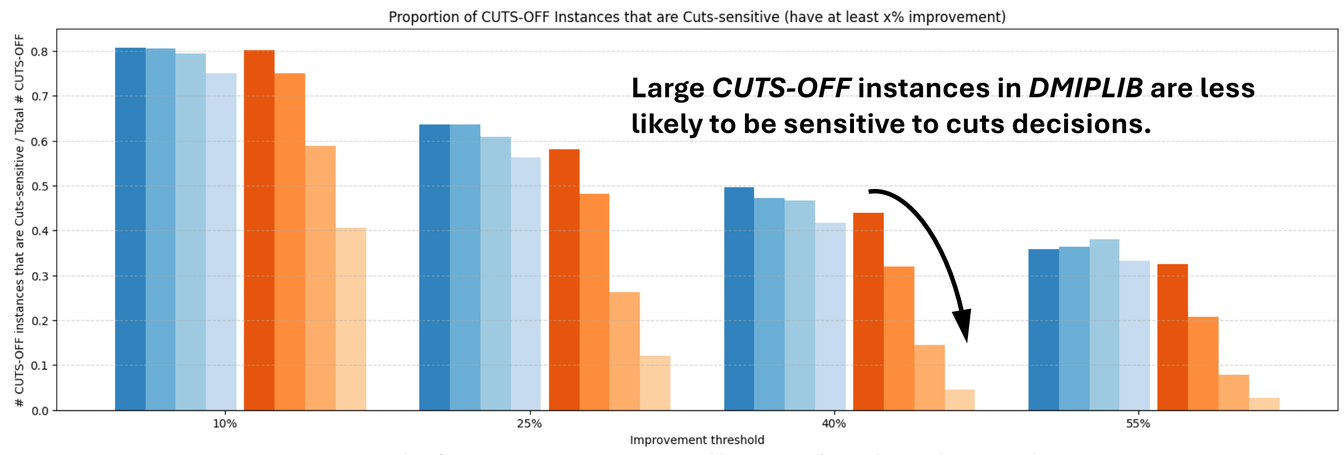
Cuts-sensitive with improvement x%:
An instance is cuts-sensitive with improvement $x\%$ if instance solves at least x% faster using best cut parameter.

$$\frac{\#\text{ CUTS-OFF, cuts-sensitive}}{\text{Total \# CUTS-OFF}}$$

$$\frac{\#\text{ CUTS-OFF, cuts-sensitive}}{\#\text{ Cuts-sensitive}}$$

Bracket (*The range of solving time within which an instance is solvable with at least one of the cut settings.*)

	[1, 10800)	[10, 10800)	[100, 10800)	[1000, 10800)
MIPLIB				
DMIPLIB				



Need: Large CUTS-OFF instances that are sensitive to cut decisions

Goal: Create scalable generators for large, high-impact CUTS-OFF instances

- Instances need to be "interesting" for learning.
- From preliminary experiments, and by folklore knowledge, certain problem classes favor *CUTS-ON* (e.g. *Set Packing* problems) and *CUTS-OFF* (e.g. *Set Covering* problems).

Idea: Can we use a mix of Set Packing and Set Covering problems to enrich the dataset?

- **Problem formulation:**

$$\min_{x,y,z} \; -\sum_{j\in\mathcal{N}} x_j + \sum_{j\in\mathcal{N}} y_j$$

Set Packing Problem	$\sum_{j\in\mathcal{S}_1} x_j \le z \quad \text{for all } \mathcal{S}_1 \in \mathcal{M}_1$
Set Covering Problem	$\sum_{j\in\mathcal{S}_2} y_j \ge 1 - z \quad \text{for all } \mathcal{S}_2 \in \mathcal{M}_2$

$$x, y \in \{0,1\}^n,\ z \in \{0,1\}$$

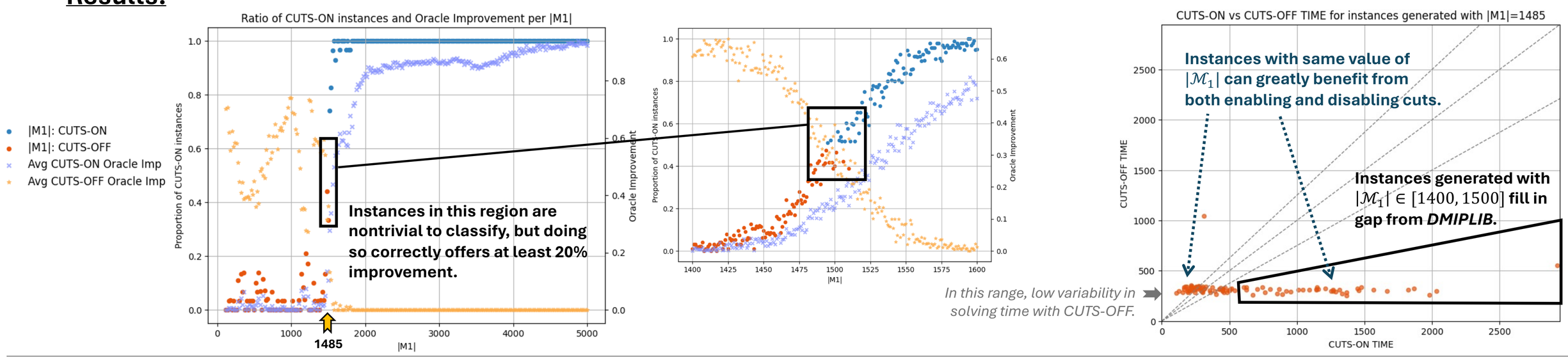
- **Metric:**

$$\textit{CUTS-ON Oracle Imp} = \frac{\textit{CUTS-OFF Time} - \textit{Oracle Time}}{\textit{CUTS-OFF Time}}$$

$$\textit{CUTS-OFF Oracle Imp} = \frac{\textit{CUTS-ON Time} - \textit{Oracle Time}}{\textit{CUTS-ON Time}}$$

(*Oracle Time*: The time required if all instances are classified correctly.)

- **Experimental setup:** $|\mathcal{N}| = 1000, |\mathcal{M}_1| + |\mathcal{M}_2| = 5000, \max(|S_1|) = \max(|S_2|) = 100$, 30 instances generated for each $|\mathcal{M}_1|$.
- **Results:**



[1] Prouvost, A., Dumouchelle, J., Scavuzzo, L., Gasse, M., Chételat, D., & Lodi, A. (2020). Ecole: A Gym-like Library for Machine Learning in Combinatorial Optimization Solvers. CoRR, abs/2011.06069.
[2] Jie Wang, Zijie Geng, Xijun Li, et al. G2MILP: Learning to Generate Mixed-Integer Linear Programming Instances for MILP Solvers. *TechRxiv*. 27 November 2023.
DOI: https://doi.org/10.36227/techrxiv.24566554.v1
[3] Huigen Ye, Yaoyang Cheng, Hua Xu, Zhiguang Cao, and Hanzhang Qin. 2025. MILPBench: A Large-scale Benchmark Test Suite for Mixed Integer Linear Programming Problems. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '25). Association for Computing Machinery, New York, NY, USA, 94–103. https://doi.org/10.1145/3712256.3726324
[4] Huang, W., Huang, T., Ferber, A.M., Dilkina, B.: Distributional MIPLIB: a Multi-Domain Library for Advancing ML-Guided MILP Methods (2024). https://arxiv.org/abs/2406.06954

This material is based upon work supported by the U.S. National Science Foundation under award No. 2441467.