To Cut Or Not To Cut

Zixuan Feng, Aleksandr M. Kazachkov, **Kausthubh Konuru** *(kausthubhkonuru@ufl.edu)* and Ambareesh P. Vaidya

Introduction & Motivation

Cutting planes (cuts) are a relaxation-tightening method for **mixed-integer programming (MIP)** problems

We focus on **globally-valid cuts** generated at the *root node* of an instance

- **CUTS-ON:** By default, modern solvers enable cuts, since they reduce average solving time over a diverse set of instances
- **CUTS-OFF:** Completely disabling cuts can cause a 50% slowdown
- **Oracle:** Taking the best of CUTS-ON and CUTS-OFF parameter settings, the **virtual best solver** (or oracle) would further improve performance

Time CUTS-ON (s)	Time CUTS-OFF (s)	Oracle (s)	Improvement (%)
74.15	113.83	54.60	**26.37**

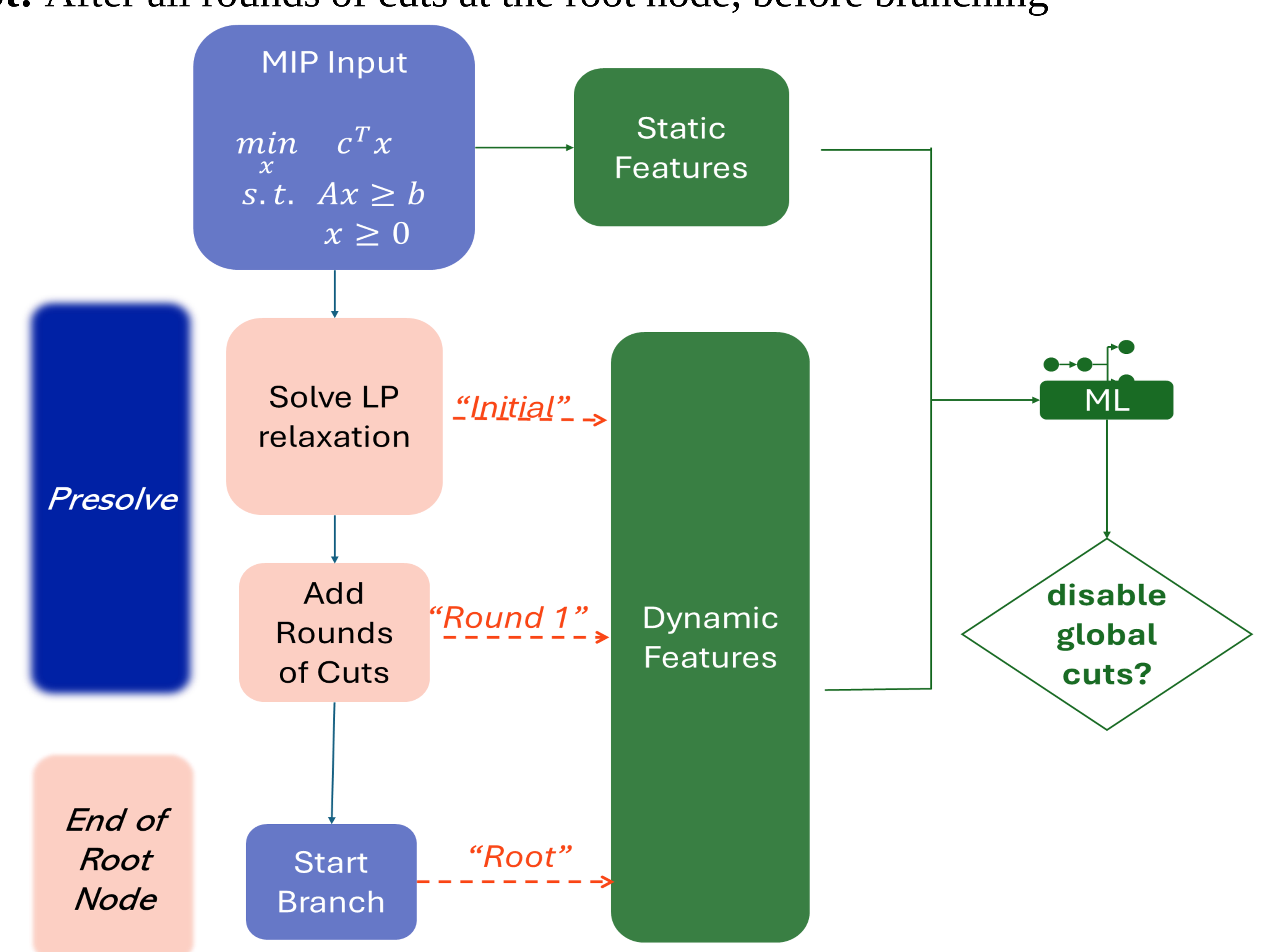
Can we predict when to use cuts based on an instance's properties?

Methodology

We adapt the methodology of Berthold, Francobaldi, Hendel (2022), who use machine learning (ML) to classify when to apply *local cuts*, generated at deeper nodes of the branch-and-bound tree.

When should we stop collecting features?
more features collected **vs** *faster solving time*

- **Initial:** After presolve and first linear programming (LP) relaxation
- **Round 1:** After one round of cuts at the root node
- **Root:** After all rounds of cuts at the root node, before branching



Results

Experimental setup:

- Each experiment uses instances from MIPLIB2017
- Use Python SCIP interface on shared cluster limited to 15gb RAM

1. Collect dynamic features (*Initial*, *Round 1*, *Root*) by solving instances with ***CUTS-ON*** and ***CUTS-OFF*** parameters with a time limit of 2 hours
2. Repeat each run with 5 random seeds for each cut setting, replicating each seed 5 times due to the shared cluster computing environment
3. Each experiment uses **extra trees (ET), random forest (RF), and support vector classifier (SVC)**

	Model	Accuracy	Precision	Recall	F1-Score	MSE Test	MSE Train
Initial	ET	**0.66**	**0.66**	**0.66**	**0.66**	**0.34**	**0.20**
	RF	0.55	0.55	0.55	0.55	0.45	0.00
	SVC	0.51	0.52	0.52	0.50	0.49	0.42
Round 1	ET	0.52	0.47	0.47	0.47	0.48	0.29
	RF	0.58	0.54	**0.53**	**0.52**	0.42	**0.00**
	SVC	**0.64**	**0.81**	**0.53**	0.45	**0.36**	0.36
Root	ET	0.73	0.62	0.51	0.46	0.27	0.22
	RF	**0.75**	**0.87**	**0.55**	**0.51**	**0.25**	**0.17**
	SVC	0.74	**0.87**	0.52	0.47	0.26	0.25

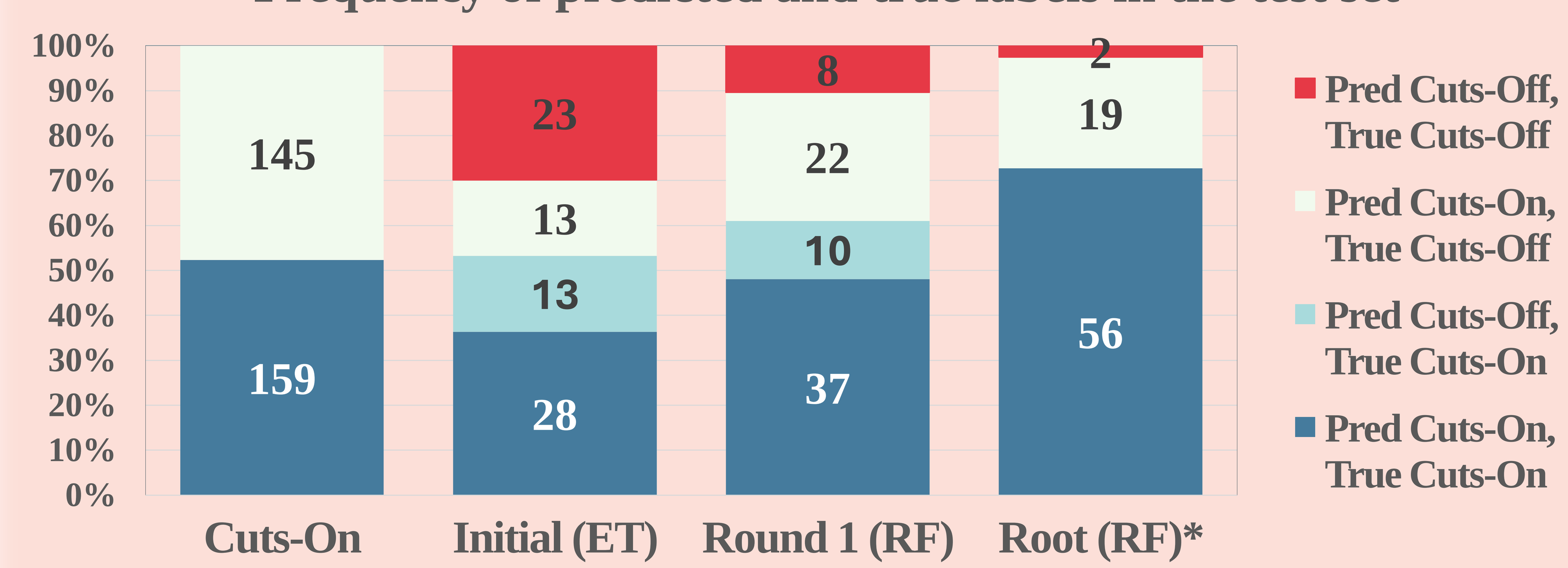
Accuracy improves for a model in each experiment as more features are added. Models with low precision suffer from favoring cuts-on for true cuts-off.

	# Instances	Metric	ET	RF	SVC	Cuts-On	Cuts-Off*	Oracle	Imp (%)
Initial	77	Time	**168.37**	175.95	219.05	**173.70**	236.42	141.18	2.96
		Node	**3,148.37**	3,545.14	5,612.22	**2,774.91**	7,388.95	2,574.35	-13.46
Round 1	77	Time	189.84	**168.93**	176.70	**175.95**	**256.24**	135.79	3.99
		Node	3,919.73	3,716.56	**2,923.76**	**2,919.27**	10,319.00	2,578.40	-27.31
Root	77	Time	182.94	**178.53**	179.43	**182.60**	326.91	152.78	2.23
		Node	3,379.96	3,221.61	**3,214.19**	**3,272.79**	10,441.38	2,796.33	1.56

**Accounts for solving time before cuts are disabled in Round 1 and Root*

In the test, "*Round 1*" has the best improvement with 1 extra feature compared to "*Initial*" and earlier stopping point compared to Root.

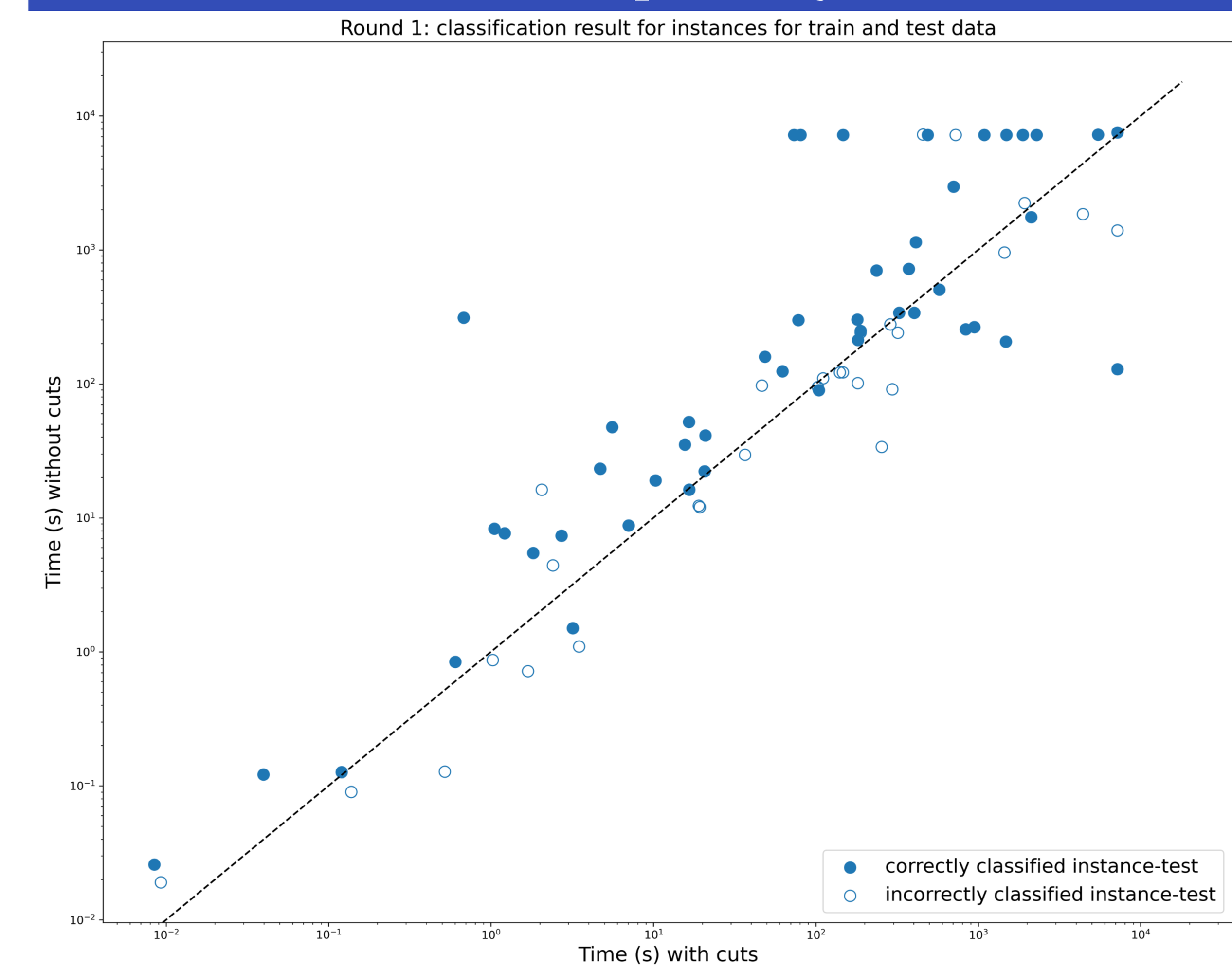
Frequency of predicted and true labels in the test set



** The frequency of cuts-off decreases in "Root" due to the later stopping point favoring cuts-on.*

Analysis

How well does the ML step classify instances that time-out?



"Round 1" has the best performance of the three experiments. Instances that cannot be solved with cuts-on can be classified and solved as cuts-off.

How well does ML improve instances that should not use cuts?

	Bracket	# Instances	RF Time	RF Nodes	Cuts-On Time	Cuts-On Nodes	Imp (%) Time	Imp (%) Nodes	Accuracy
Initial	[0, 7200]	36	100.19	2,485.93	135.73	2,559.11	26.18	**2.86**	**0.69**
	[200, 7200]	8	518.15	2,972.11	915.07	4,113.89	43.38	**27.75**	**0.75**
	[2000, 7200]	1	3,668.49	44.00	7,200.00	116.20	49.05	**62.13**	**1.00**
Round 1	[0, 7200]	30	176.22	6,449.71	249.97	5,944.99	**29.51**	-8.49	0.33
	[200, 7200]	10	683.36	23,254.26	1,554.78	27,014.02	**56.05**	13.92	0.50
	[2000, 7200]	3	1,7721.20	8,408.89	6,117.71	10,138.94	**71.05**	17.06	0.33
Root	[0, 7200]	21	457.41	10,289.39	490.57	10,799.34	6.76	4.72	0.05
	[200, 7200]	9	1,747.15	39,673.82	1,962.65	44,774.21	10.98	11.39	0.11
	[2000, 7200]	3	6,117.71	9,309.59	6,117.71	9,309.59	0.00	0.00	0.00

"Root" fails to significantly improve instances that should not use cuts as it favors predicting cuts-on for instances that are true cuts-off.

Conclusion

The augmentation of the MIP solving process using a machine learning step after the first round of cuts *(Round 1)* provides the best improvement.

- Correctly classifies instances that hit the time limit with cuts-on
- Best improvement on instances that should not use cuts

Limitations:

1. A limited number of instances
2. Too few cuts-off instances in *Root*

Future Work:

1. Try other advanced ML models: Reinforcement Learning, Deep Learning, …
2. Use **instance generators** to enrich our dataset

References

1. T. Achterberg and R. Wunderling. Mixed Integer Programming: Analyzing 12 Years of Progress, pages 449–481. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. URL https://doi.org/10.1007/978-3-642-38189-8_18.
2. T. Berthold, M. Francobaldi, and G. Hendel. Learning to use local cuts, 2022. URL https://doi.org/10.48550/arXiv.2206.11618.
3. A. Gleixner, G. Hendel, G. Gamrath, T. Achterberg, M. Bastubbe, T. Berthold, P. M. Christophel,K. Jarck, T. Koch, J. Linderoth, M. Lübbecke, H. D. Mittelmann, D. Ozyurt, T. K. Ralphs, D. Salvagnin,and Y. Shinano. MIPLIB 2017: Data-Driven Compilation of the 6th Mixed-Integer Programming Library. Math. Prog. Comp., 2021. URL https://doi.org/10.1007/s12532-020-00194-3.
4. S. Maher, M. Miltenberger, J. P. Pedroso, D. Rehfeldt, R. Schwarz, and F. Serrano. PySCIPOpt: Mathematical programming in python with the SCIP optimization suite. In G.-M. Greuel, T. Koch, P. Paule, and A. Sommese, editors, Mathematical Software – ICMS 2016, pages 301–307. Springer International Publishing, 2016. https://doi.org/10.1007/978-3-319-42432-3_37.