Improved Sample Complexity Bounds for Branch-and-Cut

Maria-Florina Balcan

Computer Science and Machine Learning Departments, Carnegie Mellon University, USA

Siddharth Prasad[1]

Computer Science Department, Carnegie Mellon University, USA

Tuomas Sandholm

Computer Science Department, Carnegie Mellon University; Optimized Markets, Inc.; Strategic Machine, Inc.; Strategy Robot, Inc., USA

Ellen Vitercik

Department of Electrical Engineering and Computer Sciences, UC Berkeley, USA

Abstract

Branch-and-cut (B&C) is a powerful algorithmic paradigm that is the backbone of all modern integer programming (IP) solvers. The main components of B&C can be tuned and tweaked in a myriad of ways. The fastest commercial IP solvers like CPLEX and Gurobi employ an array of heuristics to make decisions at every stage of B&C to reduce the solving time as much as possible, and give the user freedom to tune the multitude of parameters influencing the search through the space of feasible solutions. However, tuning the parameters that control B&C in a principled way is an inexact science with little to no formal mathematical guidelines. A rapidly growing line of work studies machine-learning approaches to speeding up the various aspects of B&C—in particular investigating whether high-performing B&C parameter configurations can be learned from a *training set* of typical IPs from the particular application at hand. Complementing the substantial number of experimental approaches using machine learning for B&C, a recent generalization theory has developed in parallel that aims to provide a rigorous theoretical foundation for how well any B&C configuration learned from training IP data will perform on new unseen IPs. In particular, this line of theoretical research provides *sample complexity guarantees* that bound how large the training set should be to ensure that *no matter how the parameters are configured* (i.e., using any approach from prior research), the average performance of branch-and-cut over the training set is close to its expected future performance. Sample complexity bounds are important because with too small a training set, learning is impossible: a configuration may have strong average performance over the training set but terrible expected performance on future IPs. If the training set is too small, then no matter how the parameters are tuned, the resulting configuration will not have reliably better performance than any default configuration. State-of-the-art parameter tuning methods have historically come without any provable guarantees, and our results fill in that gap for a wide array of tunable B&C parameters. In this paper, we expand and improve upon the existing theory to develop a wider and sharper handle on the learnability of the key components of B&C.

Our main contribution is a formalization of a general model of tree search that allows us to improve and generalize prior results on the sample complexity of tuning B&C. In this model, the algorithm repeatedly chooses a leaf node of the search tree, performs a series of actions (for example, a cutting plane to apply and a constraint to branch on), and adds children to that leaf in the search tree. The algorithm will also fathom nodes when applicable. The node and action selection are governed by *scoring rules*, which assign a real-valued score to each node and possible action. For example, a node-selection scoring rule might equal the objective value of the node's LP relaxation. We focus on general tree search with *path-wise* scoring rules. At a high level, a score of a node or action is path-wise if its value only depends on information contained along the path between the root and that node, as is often the case in B&C. Many commonly used scoring rules are path-wise including the efficacy, objective parallelism, directed cutoff distance, and integral support scoring

[1] Main student author

28th International Conference on Principles and Practice of Constraint Programming (CP 2022).
Editor: Christine Solnon; Article No. 37; pp. 37:1–37:2
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

rules, all used for cut selection by the leading open-source solver SCIP; the best-bound scoring rule for node selection; and the linear, product, and most-fractional scoring rules for variable selection using strong branching. We show how this general model of tree search captures a wide array of B&C components, including node selection, general branching constraint selection, and cutting plane selection, simultaneously. We also provide experimental evidence that, in the case of cutting plane selection, the data-dependent tuning suggested by our model can lead to dramatic reductions in the number of nodes expanded by B&C.

Our main structural insight is that for any IP, the tree search parameter space can be partitioned into a finite number of regions such that in any one region, the resulting search tree is fixed. This is in spite of the fact that the B&C search tree can be an extremely unstable function of its parameters, with minuscule changes leading to exponentially better or worse performance. By analyzing the complexity of this partition, we prove our sample complexity bound. In particular, we relate the complexity of the partition to the *pseudo-dimension* of the set of functions that measure the performance of B&C as a function of the input IP. Classic results from learning theory then allow us to translate our pseudo-dimension bound into a sample complexity guarantee, capturing the intuition that the more complex patterns one can fit (i.e., the larger the pseudo-dimension is), the more samples needed to generalize. The sample complexity bound grows linearly with the pseudo-dimension, so ideally, the pseudo-dimension will be polynomial in the size of the problem.

We show that the pseudo-dimension is only polynomial in the depth of the tree (which is, for example, at most the number of variables in the case of binary integer programming). Our bound is exponentially smaller than the pseudo-dimension bound of prior research, which grows linearly with the total number of nodes in the tree. Their results apply to any type of scoring rule, path-wise or otherwise. By taking advantage of the path-wise structure, we are able to reason inductively over the depth of the tree, leading to our exponentially improved bound. Our results recover previous results that were restricted to path-wise scoring rules for single-variable selection for branching. In contrast, we are able to handle many more of the critical components of tree search: node selection, general branching constraint selection, and cutting plane selection.

2012 ACM Subject Classification Theory of computation → Integer programming; Theory of computation → Sample complexity and generalization bounds

Keywords and phrases Automated algorithm configuration, integer programming, machine learning theory, tree search, branch-and-bound, branch-and-cut, cutting planes, sample complexity, generalization guarantees, data-driven algorithm design

Digital Object Identifier 10.4230/LIPIcs.CP.2022.37

Related Version Extend abstract of paper appearing in CP 2022 proceedings.
Full version: `https://arxiv.org/pdf/2111.11207.pdf`

Funding This material is based on work supported by the National Science Foundation under grants CCF-1733556, CCF-1910321, IIS-1901403, and SES-1919453, the ARO under award W911NF2010081, the Defense Advanced Research Projects Agency under cooperative agreement HR00112020003, a Simons Investigator Award, an AWS Machine Learning Research Award, an Amazon Research Award, a Bloomberg Research Grant, and a Microsoft Research Faculty Fellowship.