Sample Complexity of Tree Search
Configuration: Cutting Planes and Beyond

Nina Balcan Carnegie Mellon University
Siddharth Prasad Carnegie Mellon University
Tuomas Sandholm Carnegie Mellon University, Optimized Markets, Inc.,
Ellen Vitercik UC Berkeley

NeurIPS 2021
Integer programs

- Integer program (IP) in standard form:

\[
\text{Max } c \cdot x \\
\text{s.t. } Ax \leq b \\
x \in \mathbb{Z}^n
\]

- One of the most useful and widely applicable optimization techniques

Scheduling  Routing  Combinatorial auctions  Clustering
Summary of contributions

• *Cutting planes*: responsible for breakthrough speedups of IP solvers in last two decades
  – Many ways to configure how IP solvers (e.g. CPLEX, Gurobi) choose cutting planes

• Our contribution: first formal theory for using machine learning to select cutting planes
Branch-and-bound

- Powerful tree-search algorithm used to solve IPs in practice
- Uses the linear programming (LP) relaxation to do an informed search through the set of feasible integer solutions

**IP**

\[
\text{Max } c \cdot x \\
\text{s.t. } Ax \leq b \\
x \in \mathbb{Z}^n
\]

**LP relaxation**

\[
\text{Max } c \cdot x \\
\text{s.t. } Ax \leq b \\
x \in \mathbb{R}^n
\]
Branch-and-bound: branching

- Choose variable $i$ to branch on.
- Generate one subproblem with $x[i] \leq \lfloor x_{LP}[i] \rfloor$ another with $x[i] \geq \lceil x_{LP}[i] \rceil$.

Max $c \cdot x$

s.t. $Ax \leq b$

$x \in \mathbb{Z}^n$

Max $c \cdot x$

s.t. $Ax \leq b$

$x[i] \leq 2$

$x \in \mathbb{Z}^n$

Max $c \cdot x$

s.t. $Ax \leq b$

$x[i] \geq 3$

$x \in \mathbb{Z}^n$
Branch-and-bound: pruning

- Prune subtrees if
  - LP relaxation at a node is integral, infeasible, or
  - (Bounding) LP optimal worse than best feasible integer solution found so far

\[
\begin{align*}
\text{Max } & \quad c \cdot x \\
\text{s.t. } & \quad Ax \leq b \\
& \quad x[i] \leq 2 \\
& \quad x \in \mathbb{Z}^n
\end{align*}
\]
Branch-and-cut

• Branch-and-bound, but at each node may add cutting planes

• Method of getting tighter LP relaxation bounds, and thus pruning subtrees sooner
Cutting planes

- Constraint $\alpha^T x \leq \beta$ is a valid cutting plane if it does not cut off any integer feasible points.

Valid cutting planes  An invalid cutting plane
Cutting planes

- If $\alpha^T x \leq \beta$ is valid and separates the LP optimum, can speed up B&C by pruning nodes sooner.
Cutting planes

• Carefully chosen cutting planes can even achieve integrality quickly:

But finding such cutting planes is usually expensive
Cutting planes

• In the 1950s Gomory showed that any IP can be solved by a finite pure-cutting-plane algorithm
  – Highly inefficient, can require exponentially many cuts

• Nowadays IP solvers add cutting planes at various stages of B&C
Chvátal-Gomory cuts

• The Chvátal-Gomory (CG) cut parameterized by \( u \in [0,1)^m \) is the halfspace

\[
[u^T A] x \leq [u^T b]
\]

• CG cuts are valid
• Can be generated from the simplex tableau to ensure that they separate the LP optimum.
Learning to cut

Best cutting planes for airline-scheduling IPs...

...might not be useful for combinatorial-auction IPs solved by a sourcing firm
Learning to cut

If a CG cut yields small average branch-and-cut tree size over IP samples...

\[
\begin{align*}
\text{Max } & c_1 \cdot x \\
\text{s.t. } & A_1x \leq b_1 \\
& x \in \mathbb{Z}^n
\end{align*}
\]

IP 1

\[
\begin{align*}
\cdots
\end{align*}
\]

\[
\begin{align*}
\text{Max } & c_N \cdot x \\
\text{s.t. } & A_Nx \leq b_N \\
& x \in \mathbb{Z}^n
\end{align*}
\]

IP N

\sim D

...is it likely to yield a small branch-and-cut tree on a fresh IP?

\[
\begin{align*}
\text{Max } & c \cdot x \\
\text{s.t. } & Ax \leq b \\
& x \in \mathbb{Z}^n
\end{align*}
\]

\sim D
Learning to cut

• Number of samples quantified by *pseudo-dimension*
  – Measure of intrinsic complexity
  – Generalization of VC dimension to real-valued functions

• Suffices to bound pseudo-dimension of class of branch-and-cut tree-size functions parameterized by CG cuts.

• Main challenge: size of B&C tree is a complicated function of cut parameters
Sensitivity of branch-and-cut tree

- We show that small perturbations in $u$ can lead to drastically different tree sizes produced by B&C
Sensitivity of branch-and-cut tree

- We show that small perturbations in $u$ can lead to drastically different tree sizes produced by B&C.
Learning a single cut at the root

- Tree-size is a complex and highly discontinuous function of $u$
- But, it is piecewise constant

**Theorem:** For any IP $(c, A \in \mathbb{Z}^{m \times n}, b \in \mathbb{Z}^m)$, there are $O(\|A\|_{1,1} + \|b\|_1 + n)$ hyperplanes that partition $[0,1]^m$ into regions such that the tree size of B&C is constant as $u$ varies in a given region.

- This is enough to understand pseudodimension
Waves of cuts at the root

• Solvers usually add several cuts simultaneously, in waves.

• Wave 1: add cuts \( u_1^1, \ldots, u_1^k \in [0,1]^m \)

Wave 2: add cuts \( u_2^1, \ldots, u_2^k \in [0,1]^{m+k} \)

Wave w: add cuts \( u_w^1, \ldots, u_w^k \in [0,1]^{m+k(w-1)} \)
Learning waves of cuts at the root

**Theorem:** For any IP \((c, A ∈ \mathbb{Z}^{m×n}, b ∈ \mathbb{Z}^m)\) there are $O(kw2^k\|A\|_1 + 2^k\|b\|_1 + kwn)$ multivariate polynomials in $\leq k^2w^2 + mkw$ variables of degree $\leq kw$ that partition $[0,1]^m \times \cdots \times [0,1]^{k(m+k(w-1))}$ into regions such that the tree size of B&C is constant over each region.

**Proof idea (for k = 1):**

- If adding cuts $u_1, \ldots, u_w$, coefficients of wth cutting plane are degree-w polynomials in $u_1, \ldots, u_w$
- Can control the rounding aspect of CG cuts using these surfaces
Learning waves of cuts at the root

**Theorem:** The class of tree size functions parameterized by $w$ waves of $k$ CG cuts each has pseudo-dimension

$$O(mk^2w^2 \log(mkw(\alpha + \beta + n)))$$

for IPs with $\|A\|_{1,1} \leq \alpha$ and $\|b\|_1 \leq \beta$. 
Cut selection policies

• CG cut parameters may not separate LP optimum of a new unseen IP.

• Scoring rules: in practice, solvers use heuristics to choose between a pool of possible cuts.
Example of a scoring rule

**Efficacy:**

distance between cut and $x^*_\text{LP}$

\[
\text{score}_1(\alpha^T x \leq \beta) = \frac{\alpha^T x^*_\text{LP} - \beta}{\|\alpha\|_2}
\]
Learning scoring rules for CG cuts

• Given \( d \) scoring rules, learn mixture \( \mu_1 \text{score}_1 + \cdots + \mu_d \text{score}_d \) to minimize expected tree size.

• E.g., open source solver SCIP uses hardcoded weights \( \frac{3}{5} \text{score}_1 + \frac{1}{10} \text{score}_2 + \frac{1}{2} \text{score}_3 + \frac{1}{10} \text{score}_4 \)
Learning scoring rules for CG cuts

• Branch-and-cut tree size is a sensitive function

• E.g. mixture of $d = 2$ scores $\mu \cdot \text{score}_1 + (1 - \mu) \cdot \text{score}_2$
Learning scoring rules for CG cuts at the root

**Theorem**: The class of tree-size functions parameterized by \( d \) scoring-rule weights used to make \( w \) sequential CG cuts has pseudo-dimension

\[
O(dmw^2 \log(dw(\alpha + \beta + n))).
\]
General tree search

- Perform $t$ actions
- $T_j$ possible actions of type $j = 1, \ldots, t$

- Actions transition search into subsequent state
- Maximum of $\kappa$ rounds

Actions chosen using mixture of scoring rules
General tree search

**Theorem:** Let $d$ be the total number of tunable tree-search parameters. Then, the pseudodimension of the class of tree-size functions parameterized by $d$ weights is

$$O\left(\sum_{j=1}^{t} \log T_j + d \log d\right).$$

- Recovers result by Balcan et al. [ICML ’18] which was for branching/variable selection
- First guarantee that handles all aspects of branch-and-cut: node selection, variable selection, and cutting planes.
Conclusion

• We gave the first formal theory for using machine learning to select cutting planes
  – Uncovered structure in branch-and-cut tree-size
    • For waves of multiple CG cuts each
    • For mixtures of scoring rules

• Also gave the first generalization guarantees for a general form of configurable tree search