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., Strategic Machine, Inc., Strategy Robot, Inc.

Ellen Vitercik UC Berkeley

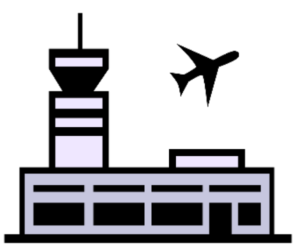
NeurIPS 2021

Integer programs

- Integer program (IP) in standard form:

$$\begin{aligned} &\text{Max } \boldsymbol{c} \cdot \boldsymbol{x} \\ &\text{s.t. } A\boldsymbol{x} \leq \boldsymbol{b} \\ &\quad\quad \boldsymbol{x} \in \mathbb{Z}^n \end{aligned}$$

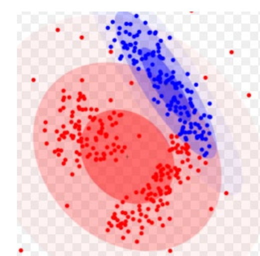
- One of the most useful and widely applicable optimization techniques



Scheduling

Routing

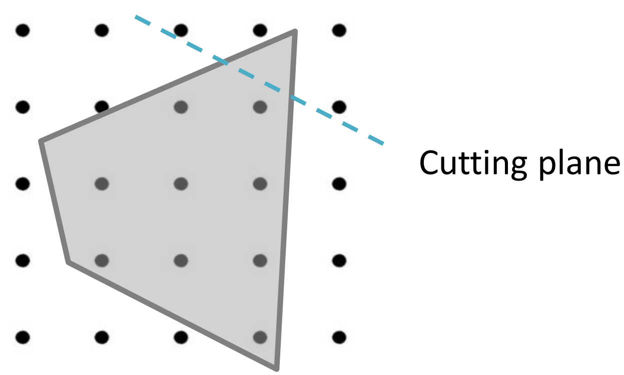
Combinatorial auctions



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

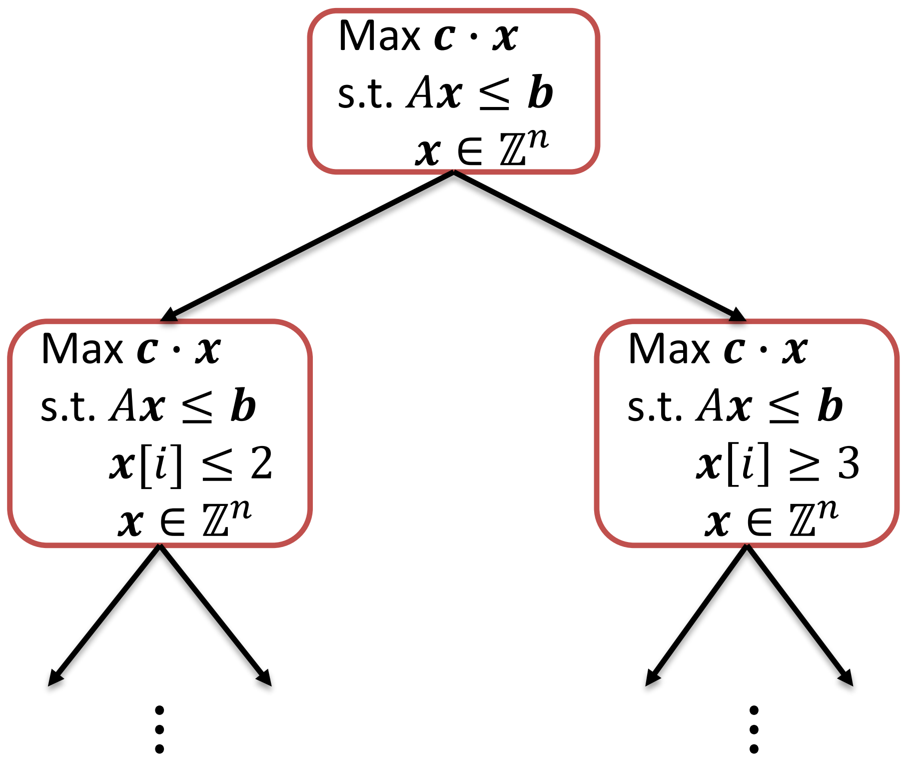
- Powerful tree-search algorithm used to solve IPs in practice

- Uses the linear programming (LP) relaxation to do an informed search through the set of feasible integer solutions

IP		LP relaxation
Max $\boldsymbol{c} \cdot \boldsymbol{x}$ s.t. $A\boldsymbol{x} \leq \boldsymbol{b}$ $\boldsymbol{x} \in \mathbb{Z}^n$	$\longrightarrow$	Max $\boldsymbol{c} \cdot \boldsymbol{x}$ s.t. $A\boldsymbol{x} \leq \boldsymbol{b}$ $\boldsymbol{x} \in \mathbb{R}^n$

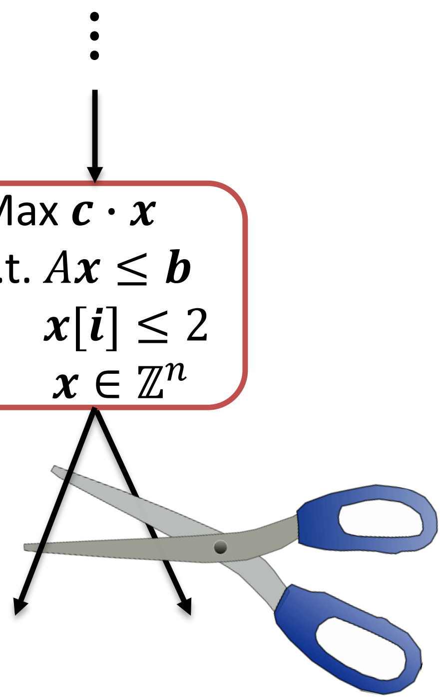
Branch-and-bound: branching

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



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

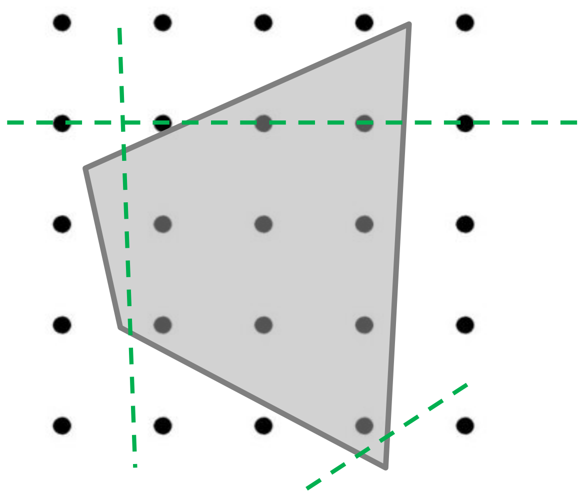


Branch-and-cut

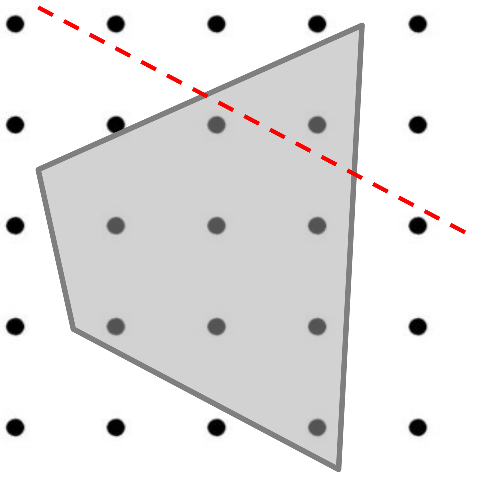
- Branch-and-bound, but at each node may add *cutting planes*
- Method of getting tighter LP relaxation bounds, and thus pruning subtrees sooner

Cutting planes

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



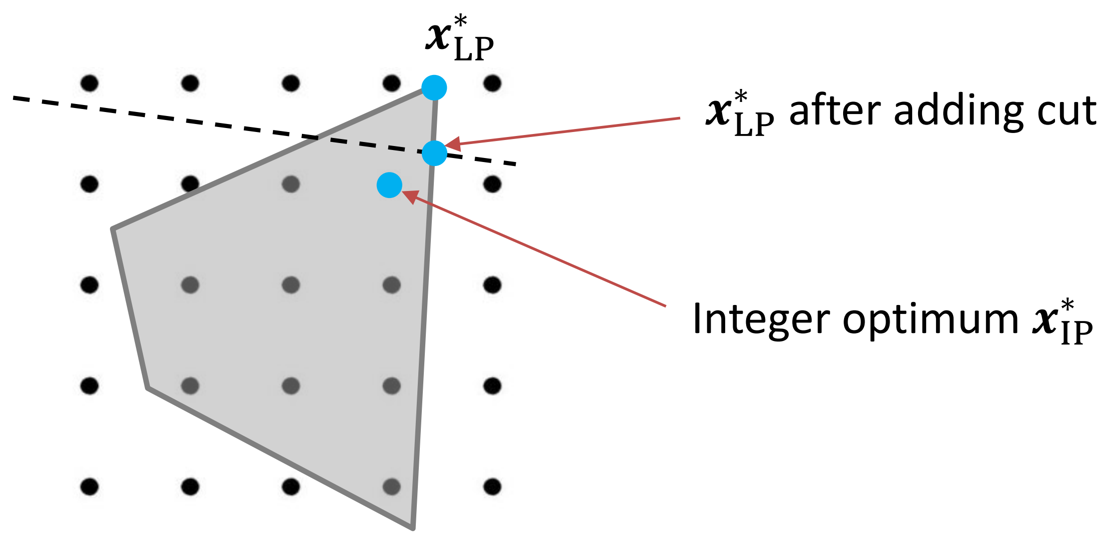
Valid cutting planes



An invalid cutting plane

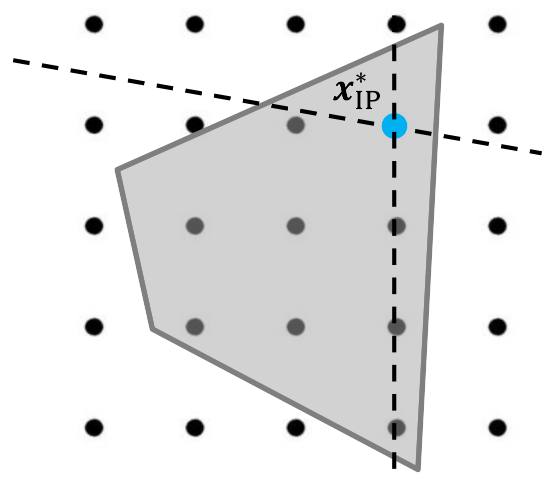
Cutting planes

- If $\boldsymbol{\alpha}^T \boldsymbol{x} \leq \beta$ 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

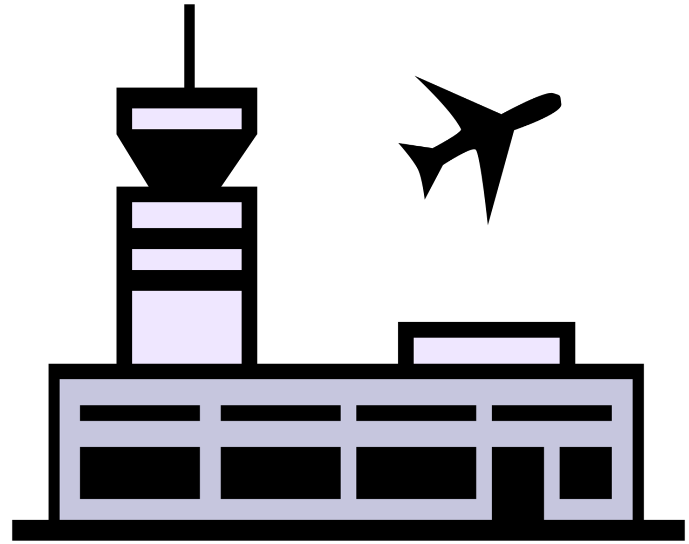
- The Chvátal-Gomory (CG) cut parameterized by $\boldsymbol{u} \in [0,1)^m$ is the halfspace

$$\lfloor \boldsymbol{u}^T A \rfloor \boldsymbol{x} \leq \lfloor \boldsymbol{u}^T \boldsymbol{b} \rfloor$$

- 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...

IP 1		IP N	
Max $\boldsymbol{c_1} \cdot \boldsymbol{x}$ s.t. $A_1\boldsymbol{x} \leq \boldsymbol{b_1}$ $\boldsymbol{x} \in \mathbb{Z}^n$	$\bullet\bullet\bullet$	Max $\boldsymbol{c}_N \cdot \boldsymbol{x}$ s.t. $A_N\boldsymbol{x} \leq \boldsymbol{b_N}$ $\boldsymbol{x} \in \mathbb{Z}^n$	$\sim D$

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

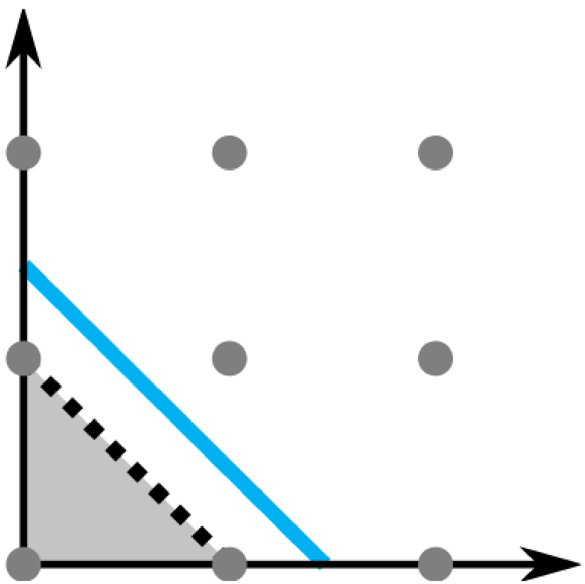
Max $\boldsymbol{c} \cdot \boldsymbol{x}$
s.t. $A\boldsymbol{x} \leq \boldsymbol{b}$
$\boldsymbol{x} \in \mathbb{Z}^n$ $\quad \sim D$

Learning to cut

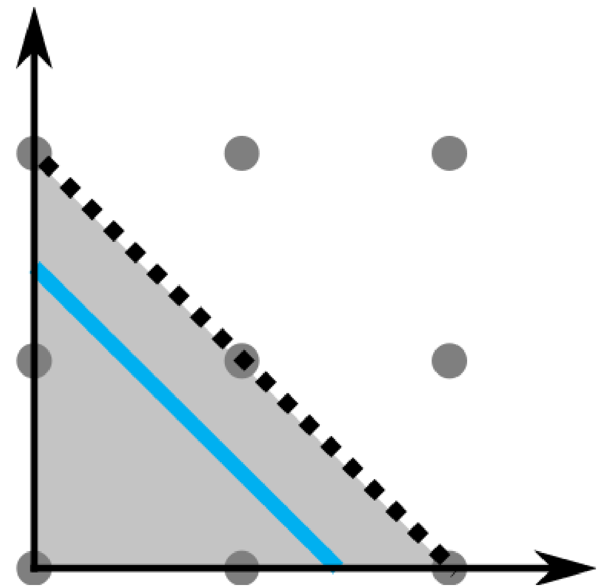
- Number of samples quantified by ***pseudo-dimension***
  - Measure of intrinsic complexity
  - Generalization of VC dimension to real-valued functions
- Suffices to bound pseudo-dimension of class of branch-and-cut tree-size functions parameterized by CG cuts.
- Main challenge: size of B&C tree is a complicated function of cut parameters

Sensitivity of branch-and-cut tree

- We show that small perturbations in $\boldsymbol{u}$ can lead to drastically different tree sizes produced by B&C



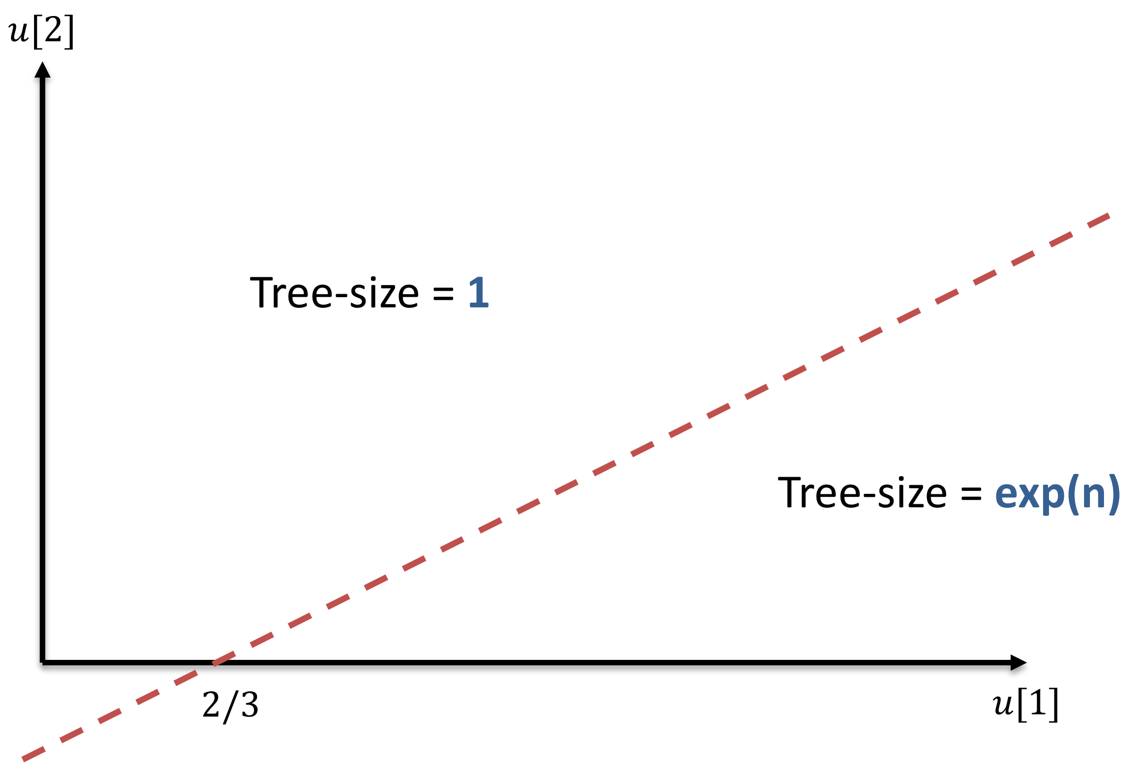
Tree-size = **1**



Tree-size = **exp(n)**

Sensitivity of branch-and-cut tree

- We show that small perturbations in $\boldsymbol{u}$ 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 $\boldsymbol{u}$
- But, it is piecewise constant

Theorem: For any IP $(\boldsymbol{c}, A \in \mathbb{Z}^{m \times n}, \boldsymbol{b} \in \mathbb{Z}^m)$, there are

$$O\left(\|A\|_{1,1} + \|\boldsymbol{b}\|_1 + n\right)$$

hyperplanes that partition $[0,1]^m$ into regions such that the tree size of B&C is constant as $\boldsymbol{u}$ 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 $\boldsymbol{u}_1^1, \dots, \boldsymbol{u}_1^k \in [0,1]^m$

  Wave 2: add cuts $\boldsymbol{u}_2^1, \dots, \boldsymbol{u}_2^k \in [0,1]^{m+k}$

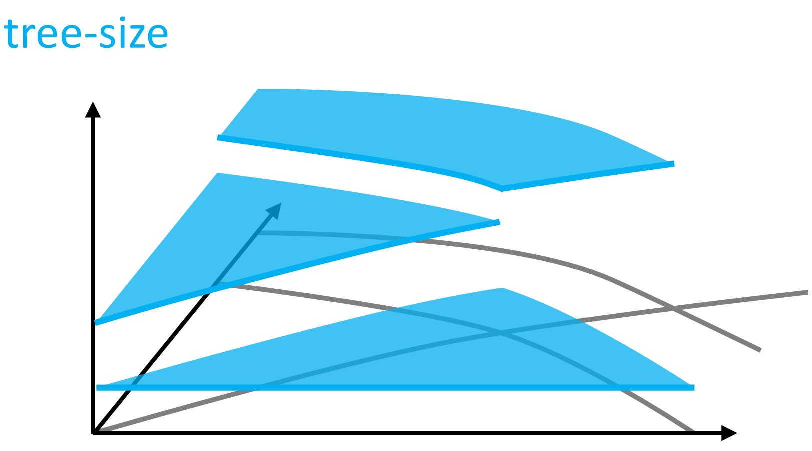
  Wave w: add cuts $\boldsymbol{u}_w^1, \dots, \boldsymbol{u}_w^k \in [0,1]^{m+k(w-1)}$

Learning waves of cuts at the root

Theorem: For any IP $(\boldsymbol{c}, A \in \mathbb{Z}^{m\times n}, \boldsymbol{b} \in \mathbb{Z}^m)$ there are

$$O\left(kw2^{kw}\|A\|_{1,1} + 2^{kw}\|\boldsymbol{b}\|_1 + kwn\right)$$

multivariate polynomials in $\leq k^2w^2 + mkw$ variables of degree $\leq kw$ that partition $[0,1]^{mk} \times \cdots \times [0,1]^{k(m+k(w-1))}$ into regions such that the tree size of B&C is constant over each region.



Proof idea (for k = 1):

- If adding cuts $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_w$, coefficients of wth cutting plane are degree-w polynomials in $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_w$
- 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

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

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

Cut selection policies

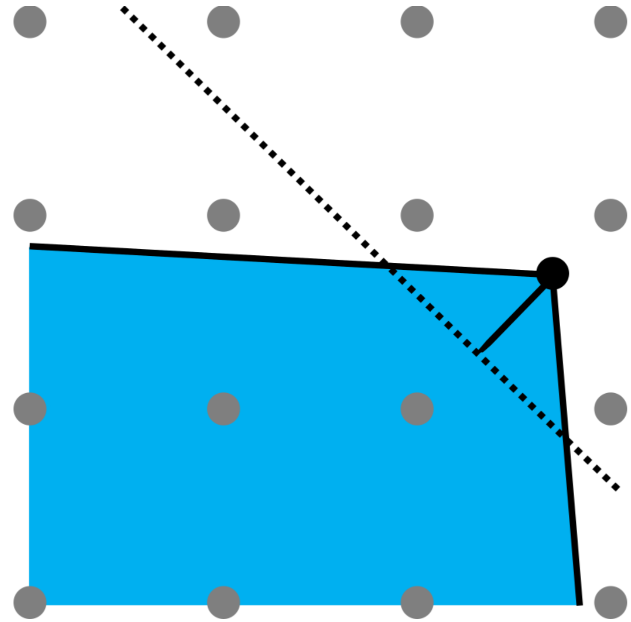
- CG cut parameters may not separate LP optimum of a new unseen IP.

- 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 $\boldsymbol{x}_{\mathrm{LP}}^*$



$$\mathrm{score}_1(\boldsymbol{\alpha}^T\boldsymbol{x} \leq \beta) = \frac{\boldsymbol{\alpha}^T\boldsymbol{x}_{\mathrm{LP}}^* - \beta}{\|\boldsymbol{\alpha}\|_2}$$

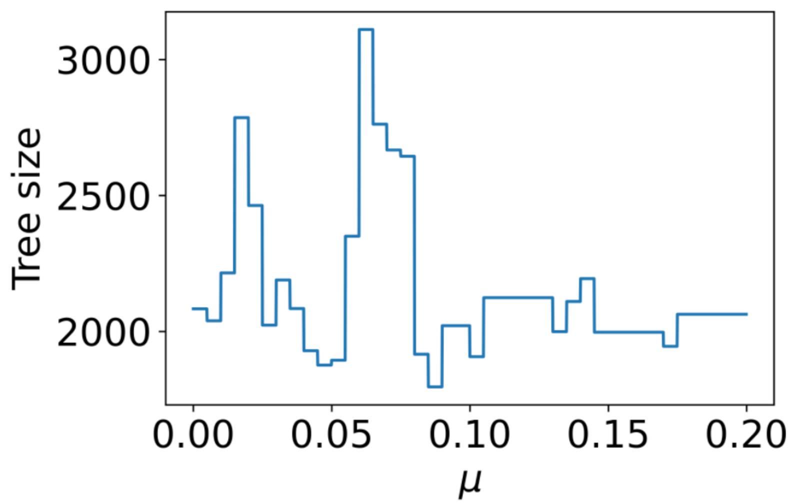
Learning scoring rules for CG cuts

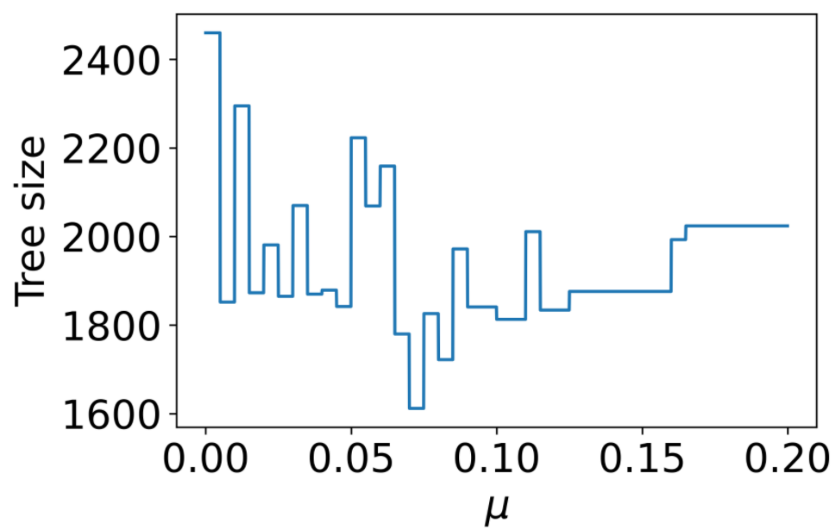
- Given d scoring rules,
  learn mixture $\mu_1 \mathrm{score}_1 + \cdots + \mu_d \mathrm{score}_d$
  to minimize expected tree size.

- E.g., open source solver SCIP uses hardcoded weights
  $\frac{3}{5}\mathrm{score}_1 + \frac{1}{10}\mathrm{score}_2 + \frac{1}{2}\mathrm{score}_3 + \frac{1}{10}\mathrm{score}_4$

Learning scoring rules for CG cuts

- Branch-and-cut tree size is a sensitive function
- E.g. mixture of $d = 2$ scores $\mu \cdot \text{score}_1 + (1 - \mu) \cdot \text{score}_2$



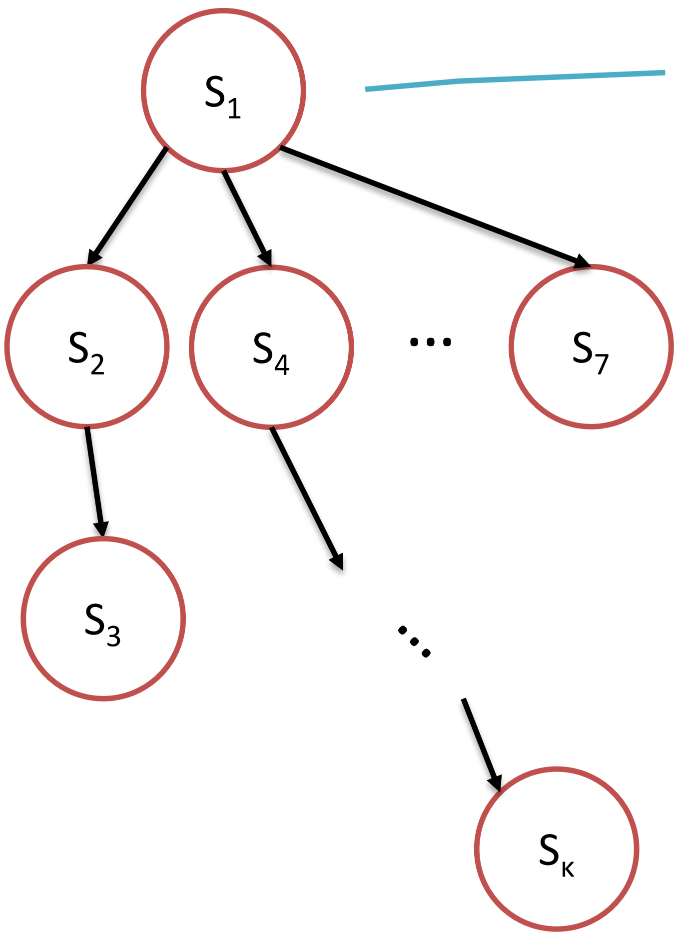


Learning scoring rules for CG cuts at the root

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

$$O\left(dmw^2 \log\left(dw(\alpha + \beta + n)\right)\right).$$

General tree search



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

- Actions transition search into subsequent state

- Maximum of κ 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

Number of rounds   Number of possible actions of type j

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

- 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