Learning to Branch

Ellen Vitercik
Automated Algorithms Reading Group
April 19, 2018

Joint work with Nina Balcan, Travis Dick, and Tuomas Sandholm
Mixed Integer Programs (MIPs)

maximize $c \cdot x$
subject to $Ax = b$
\hspace{1cm} $x_i \in \{0,1\}, \forall i \in I$
Facility location problems can be formulated as MIPs.
Clustering problems can be formulated as MIPs.
Binary classification problems can be formulated as MIPs.
Branch and Bound (B&B)

The most widely-used algorithm for MIP-solving (CPLEX, Gurobi). CPLEX has a 221-page manual describing 135 parameters:

“You may need to experiment.”
B&B in the Real World

A delivery company routes trucks through Pittsburgh on a daily basis. Demand changes every day, so hundreds of similar optimizations must be solved.

Using this set of typical problem instances, can we learn the best parameters for this domain?
How can I use the set of samples to find B&B parameters that are best for my application domain?
This model has been studied in applied communities (e.g. [Hutter et al. ‘09]).
Model

This model has been studied from a theoretical perspective [Gupta and Roughgarden ‘16, Balcan et al., ‘17].
1. Fix a set of B&B parameters to optimize
2. Receive sample problems from an unknown distribution
3. Find parameters with the best performance on the samples

“Best” could mean smallest search tree, for example
Questions to Address

How should we find parameters with the best performance on the set of samples?

\[ \{A^{(1)}, b^{(1)}, c^{(1)}\} \quad \text{and} \quad \{A^{(2)}, b^{(2)}, c^{(2)}\} \]
Questions to Address

How should we find parameters with the best performance on the set of samples?

The resulting parameters have high performance over the samples, but will they have high performance in expectation over the unknown distribution?
Outline

1. Branch-and-Bound
   i. Algorithm Overview
   ii. Variable Selection Policies
2. Learning Algorithms
   i. First-try: Discretization
   ii. Our Approach
3. Generalization Guarantees
4. Experiments
5. Conclusion and Future Directions
\[ \text{max } (40, 60, 10, 10, 30, 20, 60) \cdot x \]
\[ \text{s.t. } (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100 \]
\[ x \in \{0,1\}^7 \]
max \ (40, 60, 10, 10, 30, 20, 60) \cdot x
s.t. \ (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100
\ x \in \{0,1\}^7

\begin{array}{c}
(\frac{1}{2}, 1, 0, 0, 0, 0, 1) \\
140
\end{array}
1. Choose a leaf of the search tree

\[
\begin{align*}
\text{max} & \quad (40, 60, 10, 10, 30, 20, 60) \cdot x \\
\text{s.t.} & \quad (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100 \\
& \quad x \in \{0, 1\}^7
\end{align*}
\]
B&B

1. Choose a leaf of the search tree
2. Branch on a variable

\[
\begin{align*}
\text{max} \quad & (40, 60, 10, 10, 30, 20, 60) \cdot x \\
\text{s.t.} \quad & (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100 \\
& x \in \{0, 1\}^7
\end{align*}
\]

\[
(x_1 = 0, x_2 = 1, x_3 = 0, x_4 = \frac{1}{4}, x_5 = 1) \\
135
\]

\[
(x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 0, x_5 = 1) \\
140
\]

\[
(x_1 = 1, x_2 = \frac{3}{5}, x_3 = 0, x_4 = 0, x_5 = 1) \\
136
\]
B&B

1. Choose a leaf of the search tree
2. Branch on a variable
3. Fathom a node if:
   i. The optimal solution to its LP relaxation satisfies the constraints of the original problem
   ii. Its LP relaxation is infeasible
   iii. The optimal solution to its LP relaxation is no better than the best-known feasible solution

\[
\begin{align*}
\text{max } & \quad (40, 60, 10, 10, 30, 20, 60) \cdot x \\
\text{s.t. } & \quad (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100 \\
& \quad x \in \{0,1\}^7
\end{align*}
\]
1. Choose a leaf of the search tree
2. Branch on a variable
3. Fathom a node if:
   i. The optimal solution to its LP relaxation satisfies the constraints of the original problem
   ii. Its LP relaxation is infeasible
   iii. The optimal solution to its LP relaxation is no better than the best-known feasible solution

\[
\text{max } (40, 60, 10, 10, 30, 20, 60) \cdot x \\
\text{s.t. } (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100 \\
x \in \{0,1\}^7
\]
1. Choose a leaf of the search tree
2. Branch on a variable
3. Fathom a node if:
   i. The optimal solution to its LP relaxation satisfies the constraints of the original problem
   ii. Its LP relaxation is infeasible
   iii. The optimal solution to its LP relaxation is no better than the best-known feasible solution

max \( (40, 60, 10, 10, 30, 20, 60) \cdot x \)
\[\text{s.t. } (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100\]
\( x \in \{0, 1\}^7 \)

\[
\begin{align*}
\text{max} & \quad (40, 60, 10, 10, 30, 20, 60) \cdot x \\
\text{s.t.} & \quad (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100 \\
& \quad x \in \{0, 1\}^7
\end{align*}
\]
1. Choose a leaf of the search tree
2. Branch on a variable
3. Fathom a node if:
   i. The optimal solution to its LP relaxation satisfies the constraints of the original problem
   ii. Its LP relaxation is infeasible
   iii. The optimal solution to its LP relaxation is no better than the best-known feasible solution

\[
\begin{align*}
\text{max } & \quad (40, 60, 10, 10, 30, 20, 60) \cdot x \\
\text{s.t. } & \quad (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100 \\
& \quad x \in \{0,1\}^7
\end{align*}
\]
1. Choose a leaf of the search tree

2. **Branch on a variable**

3. Fathom a node if:
   i. The optimal solution to its LP relaxation satisfies the constraints of the original problem
   ii. Its LP relaxation is infeasible
   iii. The optimal solution to its LP relaxation is no better than the best-known feasible solution

**Example:**

\[
\begin{align*}
\text{max} & \quad (40, 60, 10, 10, 30, 20, 60) \cdot x \\
\text{s.t.} & \quad (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100 \\
& \quad x \in \{0,1\}^7
\end{align*}
\]
1. Choose a leaf of the search tree
2. Branch on a variable
3. Fathom a node if:
   i. The optimal solution to its LP relaxation satisfies the constraints of the original problem
   ii. Its LP relaxation is infeasible
   iii. The optimal solution to its LP relaxation is no better than the best-known feasible solution

\[
\begin{align*}
\text{max} & \quad (40, 60, 10, 10, 30, 20, 60) \cdot x \\
\text{s.t.} & \quad (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100 \\
& \quad x \in \{0,1\}^7
\end{align*}
\]
1. Choose a leaf of the search tree
2. Branch on a variable
3. Fathom a node if:
   i. The optimal solution to its LP relaxation satisfies the constraints of the original problem
   ii. Its LP relaxation is infeasible
   iii. The optimal solution to its LP relaxation is no better than the best-known feasible solution

\[
\begin{align*}
\text{max} & \quad (40, 60, 10, 10, 30, 20, 60) \cdot x \\
\text{s.t.} & \quad (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100 \\
& \quad x \in \{0, 1\}^7
\end{align*}
\]

\[
\left(\frac{1}{2}, 1, 0, 0, 0, 0, 1\right) \quad 140
\]

\[
\left(1, \frac{3}{5}, 0, 0, 0, 0, 1\right) \quad 136
\]

\[
\begin{align*}
(0, 1, 0, 1, 0, \frac{1}{4}, 1) & \quad 135 \\
(0, 1, \frac{1}{3}, 1, 0, 0, 1) & \quad 133 \frac{1}{3} \\
(0, \frac{3}{5}, 0, 0, 0, 1, 1) & \quad 116 \\
(1, 0, 0, 1, 0, \frac{1}{2}, 1) & \quad 120 \\
(1, 1, 0, 0, 0, 0, \frac{1}{3}) & \quad 120
\end{align*}
\]

\[
\begin{align*}
x_1 = 0 & \quad x_1 = 1 \\
x_6 = 0 & \quad x_6 = 1 \\
x_2 = 0 & \quad x_2 = 1 \\
x_3 = 0 & \quad x_3 = 1
\end{align*}
\]
B&B

1. Choose a leaf of the search tree
2. Branch on a variable
3. **Fathom a node if:**
   i. The optimal solution to its LP relaxation satisfies the constraints of the original problem
   ii. Its LP relaxation is infeasible
   iii. The optimal solution to its LP relaxation is no better than the best-known feasible solution

### Objective Function

\[
\text{max } (40, 60, 10, 10, 30, 20, 60) \cdot x
\]

### Constraints

\[
(40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100
\]

\[
x \in \{0,1\}^7
\]
1. Choose a leaf of the search tree
2. Branch on a variable
3. **Fathom a node if:**
   i. The optimal solution to its LP relaxation satisfies the constraints of the original problem
   ii. Its LP relaxation is infeasible
   iii. The optimal solution to its LP relaxation is no better than the best-known feasible solution

\[
\text{max } (40, 60, 10, 10, 30, 20, 60) \cdot x \\
\text{s.t. } (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100 \\
x \in \{0,1\}^7
\]
1. Choose a leaf of the search tree
2. **Branch on a variable**
3. Fathom a node if:
   i. The optimal solution to its LP relaxation satisfies the constraints of the original problem
   ii. Its LP relaxation is infeasible
   iii. The optimal solution to its LP relaxation is no better than the best-known feasible solution

**This talk:** How do we choose which variable? (Assume every other aspect of B&B, such as the node selection policy, is fixed.)
Prior Work

These works are purely empirical, whereas we are the first to give provable guarantees for tree search algorithm selection.

- There are many works on ML for aspects of B&B other than variable selection [He et al., 2014; Sabharwal et al., 2017; Hutter et al., 2009; Kruber et al., 2017; Khalil et al., 2017].
- Khalil et al. (2016) study variable selection policies:
  - They split a single MIP into a training and test set. The beginning of B&B is the training phase: the learning algorithm assigns features to each node and observes the variable selected by a classic branching policy. It then learns a faster branching policy meant to simulate the classic branching policy.
  - Alvarez et al. (2017) study a similar problem, although in their work, the feature vectors in the training set describe nodes from multiple MIPs.
- Di Liberto et al. (2016) study the use of ML for dynamically switching between branching heuristics.
- Xia & Yap (2018) and Balafrej et al. (2015) study ML for CSP tree search
- Some papers study algorithm selection from a purely theoretical perspective, whereas ours includes both theory and experiments [Gupta and Roughgarden, 2017; Balcan et al. 2017].
Outline

1. Branch-and-Bound
   i. Algorithm Overview
   ii. Variable Selection Policies
2. Learning Algorithms
   i. First-try: Discretization
   ii. Our Approach
3. Generalization Guarantees
4. Experiments
5. Conclusion and Future Directions
For a MIP instance $Q$:
- Let $c_Q$ be the objective value of its LP relaxation

Example.

\[
\begin{align*}
\text{max} & \quad (40, 60, 10, 10, 30, 20, 60) \cdot x \\
\text{s.t.} & \quad (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100 \\
& \quad x \in \{0, 1\}^7 \\
\end{align*}
\]

\[
\begin{align*}
\left(\frac{1}{2}, 1, 0, 0, 0, 0, 1\right) \\
140
\end{align*}
\]

$c_Q$
Variable Selection Policies

For a MIP instance $Q$:

- Let $c_Q$ be the objective value of its LP relaxation
- Let $Q_i^-$ be $Q$ with $x_i$ set to 0, and let $Q_i^+$ be $Q$ with $x_i$ set to 1

Example.

\[\begin{align*}
\text{max} & \quad (40, 60, 10, 10, 30, 20, 60) \cdot x \\
\text{s.t.} & \quad (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100 \\
& \quad x \in \{0,1\}^7
\end{align*}\]
Variable Selection Policies

**The linear rule (parameterized by \( \mu \)) [Linderoth & Savelsbergh, 1999]**

Branch on the fractional variable \( x_i \) that maximizes:

\[
\text{score}(Q,i) = \mu \min \left\{ c_Q - c_{Q_i}, c_Q - c_{Q_i^+} \right\} + (1 - \mu) \max \left\{ c_Q - c_{Q_i}, c_Q - c_{Q_i^+} \right\}
\]

### Example

\[
\begin{align*}
Q & \quad \text{max} \quad (40, 60, 10, 30, 20, 60) \cdot x \\
& \quad \text{s.t.} \quad (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100 \\
& \quad x \in \{0,1\}^7
\end{align*}
\]

### Scores

- \( c_{Q_1} = 140 \)
- \( c_{Q_1^-} = 135 \)
- \( c_{Q_1^+} = 136 \)

---

\[
\begin{align*}
\frac{1}{2}, 1, 0, 0, 0, 0, 1 & \quad c_Q \\
(0, 1, 0, 1, 0, 1, 4) & \quad c_{Q_1^-} \\
(1, 3, 0, 0, 0, 0, 1) & \quad c_{Q_1^+}
\end{align*}
\]
Variable Selection Policies

<table>
<thead>
<tr>
<th>The linear rule (parameterized by $\mu$) [Linderoth &amp; Savelsbergh, 1999]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branch on the fractional variable $x_i$ that maximizes:</td>
</tr>
<tr>
<td>$\text{score}(Q, i) = \mu \min{c_Q - c_{Q_i^-}, c_Q - c_{Q_i^+}} + (1 - \mu) \max{c_Q - c_{Q_i^-}, c_Q - c_{Q_i^+}}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The (simplified) product rule [Achterberg, 2009]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branch on the fractional variable $x_i$ that maximizes:</td>
</tr>
<tr>
<td>$\text{score}(Q, i) = (c_Q - c_{Q_i^-}) \cdot (c_Q - c_{Q_i^+})$</td>
</tr>
</tbody>
</table>

And many more...
Parameterized Variable Selection Policies

We take \( d \) scoring rules \( \text{score}_1, \ldots, \text{score}_d \) and **learn** the best convex combination \( \mu_1 \text{score}_1 + \cdots + \mu_d \text{score}_d \).

**Our parameterized rule**

<table>
<thead>
<tr>
<th>Branch on the fractional variable ( x_i ) that maximizes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{score}(Q,i) = \mu_1 \text{score}_1(Q,i) + \cdots + \mu_d \text{score}_d(Q,i) )</td>
</tr>
</tbody>
</table>
Application-Specific Distribution

Model

\[
\{A^{(1)}, b^{(1)}, c^{(1)}\}, \ldots, \{A^{(m)}, b^{(m)}, c^{(m)}\}
\]

B&B parameters

Algorithm Designer

How can I use the set of samples to find B&B parameters that are best for my application domain?
How can I use the set of samples to find $\mu_1, \ldots, \mu_d$ that are best for my application domain?
Outline

1. Branch-and-Bound
   i. Algorithm Overview
   ii. Variable Selection Policies

2. Learning Algorithms
   i. First-try: Discretization
   ii. Our Approach

3. Generalization Guarantees

4. Experiments

5. Conclusion and Future Directions
First Try: Discretization

1. Discretize the parameter space
2. Receive sample problems from an unknown distribution
3. Find parameters in the discretized set with the best performance on the samples

Average tree size
First Try: Discretization

This has been prior work’s approach [e.g., Achterberg (2009)].

Average tree size
Discretization Gone Wrong

Average tree size

\[ \mu \]
Discretization Gone Wrong

Average tree size

This can actually happen!
Worst-Case Distributions for Discretization

Let:

\[
\text{score}_1(Q,i) = \min \left\{ c_Q - c_{Q^-}, c_Q - c_{Q^+} \right\}
\]

\[
\text{score}_2(Q,i) = \max \left\{ c_Q - c_{Q^-}, c_Q - c_{Q^+} \right\}
\]

Example.

\[
\max \quad (40, 60, 10, 10, 30, 20, 60) \cdot x \\
\text{s.t.} \quad (40, 50, 30, 10, 10, 40, 30) \cdot x \leq 100 \\
x \in \{0,1\}^7
\]
Worst-Case Distributions for Discretization

Let:
\[
\begin{align*}
\text{score}_1(Q, i) &= \min \left\{ c_Q - c_{Q^-}, c_Q - c_{Q^+} \right\} \\
\text{score}_2(Q, i) &= \max \left\{ c_Q - c_{Q^-}, c_Q - c_{Q^+} \right\}
\end{align*}
\]

Our parameterized rule

Branch on the fractional variable \(x_i\) that maximizes:
\[
\text{score}(Q, i) = \mu \cdot \text{score}_1(Q, i) + (1 - \mu) \cdot \text{score}_2(Q, i)
\]

This is the linear rule [Linderoth & Savelsbergh, 1999]
Worst-Case Distributions for Discretization

Let: \( \text{score}_1(Q, i) = \min \{ c_Q - c_{Q^-}, c_Q - c_{Q^+} \} \)

\( \text{score}_2(Q, i) = \max \{ c_Q - c_{Q^-}, c_Q - c_{Q^+} \} \)

**Theorem** (informal)

For any discretization of \([0, 1]\), there exists a distribution over MIPs such that for any parameter \(\mu\) in the discretization, the expected size of the tree B&B builds using the scoring rule \(\mu \cdot \text{score}_1 + (1 - \mu) \cdot \text{score}_2\) is exponential. Yet, there exists an infinite number of parameters such that the tree size is constant with probability 1.
Worst-Case Distributions for Discretization

**Theorem (informal)**
For any discretization of [0, 1], there exists a distribution over MIPs such that for any parameter $\mu$ in the discretization, the expected size of the tree B&B builds using the scoring rule $\mu \cdot \text{score}_1 + (1 - \mu) \cdot \text{score}_2$ is exponential. Yet, there exists an infinite number of parameters such that the tree size is constant with probability 1.

**Proof idea:**
Jeroslow [1974] proved B&B **always builds an exponential-sized tree** on this MIP (no matter the variable selection policy!), so long as $n$ is odd:

\[
\begin{align*}
\text{maximize} & \quad x_1 \\
\text{subject to} & \quad 2 \sum_{i=1}^{n} x_i = n, \\
& \quad x \in \{0,1\}^n.
\end{align*}
\]
Proof idea (continued): Fix an arbitrary discretization $\mu$

- We create 2 MIPs with tree-size plots $\mathbf{1}$ and $\mathbf{2}$ and place equal probability mass on them.
- These two MIPs combine a hard version of Jeroslow’s instance on $n - 3$ variables $\{x_1, \ldots, x_{n-3}\}$ and an easy version on 3 variables $\{x_{n-2}, x_{n-1}, x_n\}$.
- B&B only needs to determine that one of these problems is infeasible to terminate.
  - If it branches on $\{x_{n-2}, x_{n-1}, x_n\}$ first, it will terminate upon making a small tree.
  - If it branches on $\{x_1, \ldots, x_{n-3}\}$ first, it will create a tree with exponential size.
- The challenge is to design an objective function that enforces this behavior.
Proof idea (continued): Fix an arbitrary discretization $\mu$

- We create 2 MIPs with tree-size plots $\textcircled{1}$ and $\textcircled{2}$ and place equal probability mass on them.
- These two MIPs combine a hard version of Jeroslow’s instance on $n - 3$ variables $\{x_1, ..., x_{n-3}\}$ and an easy version on 3 variables $\{x_{n-2}, x_{n-1}, x_n\}$.
- B&B only needs to determine that one of these problems is infeasible to terminate.
  - If it branches on $\{x_{n-2}, x_{n-1}, x_n\}$ first, it will terminate upon making a small tree.
  - If it branches on $\{x_1, ..., x_{n-3}\}$ first, it will create a tree with exponential size.
- The challenge is to design an objective function that enforces this behavior.

\textbf{Worst-Case Distributions for Discretization}
Worst-Case Distributions for Discretization

**Proof idea** (continued):
For an example’s sake, suppose $n = 8$. Consider the MIP:

maximize $\left( 1, 2, 3, 4, 5, 0, \frac{3}{2}, 3 - \frac{1}{2a} \right) \cdot x$

subject to

$(2 \ 2 \ 2 \ 2 \ 0 \ 0 \ 0) x = (5)$

$x \in \{0, 1\}^8$
Worst-Case Distributions for Discretization

Proof idea (continued):
For an example’s sake, suppose \( n = 8 \). Consider the MIP:

maximize \( \begin{pmatrix} 1, 2, 3, 4, 5, 0, \frac{3}{2}, 3 - \frac{1}{2a} \end{pmatrix} \cdot x \)

subject to \( \begin{pmatrix} 2 & 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \end{pmatrix} x = \begin{pmatrix} 5 \\ 3 \end{pmatrix} \)

\( x \in \{0, 1\}^8 \)

Big version of Jeroslow’s instance.
Proof idea (continued):
For an example’s sake, suppose $n = 8$. Consider the MIP:

$$\begin{align*}
\text{maximize} & \quad \left(1, 2, 3, 4, 5, \frac{3}{2}, 3 - \frac{1}{2a}\right) \cdot x \\
\text{subject to} & \quad \begin{pmatrix} 2 & 2 & 2 & 2 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \end{pmatrix} x = \begin{pmatrix} 5 \\ 3 \end{pmatrix} \\
x & \in \{0, 1\}^8
\end{align*}$$

Small version of Jeroslow’s instance.
Worst-Case Distributions for Discretization

Proof idea (continued):

For an example’s sake, suppose \( n = 8 \). Consider the MIP:

maximize \( (1, 2, 3, 4, 5, 0, \frac{3}{2}, 3 - \frac{1}{2a}) \cdot x \)

subject to \( \begin{pmatrix}
2 & 2 & 2 & 2 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 2 & 2
\end{pmatrix} x = \begin{pmatrix} 5 \\ 3 \end{pmatrix} \)

\( x \in \{0, 1\}^8 \)

The solution to the LP relaxation has \( (x_1, ..., x_5) = \left(0, 0, \frac{1}{2}, 1, 1\right) \)
Worst-Case Distributions for Discretization

**Proof idea** (continued):
For an example’s sake, suppose \( n = 8 \). Consider the MIP:

\[
\begin{align*}
\text{maximize} & \quad (1, 2, 3, 4, 5, 0, \frac{3}{2}, 3 - \frac{1}{2a}) \cdot x \\
\text{subject to} & \quad (2, 2, 2, 2, 2, 0, 0, 0) \cdot x = \left(\begin{array}{c} 5 \\ 3 \end{array}\right) \\
x & \in \{0, 1\}^8
\end{align*}
\]

The solution to the LP relaxation has \((x_1, \ldots, x_8) = (0, 0, \frac{1}{2}, 1, 1)\) and \((x_6, x_7, x_8) = (0, \frac{1}{2}, 1)\).
Proof idea (continued):
For an example’s sake, suppose $n = 8$. Consider the MIP:

$$\text{maximize } (1, 2, 3, 4, 5, 0, \frac{3}{2}, 3 - \frac{1}{2a}) \cdot x$$

$$\text{subject to } \begin{pmatrix} 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \end{pmatrix} x = \begin{pmatrix} 5 \\ 3 \end{pmatrix}$$

$x \in \{0, 1\}^8$

The solution to the LP relaxation has $(x_1, \ldots, x_5) = (0, 0, \frac{1}{2}, 1, 1)$ and $(x_6, x_7, x_8) = (0, \frac{1}{2}, 1)$.

Will either branch on $x_3$ or $x_7$. 

Worst-Case Distributions for Discretization
Proof idea (continued):
For an example’s sake, suppose \( n = 8 \). Consider the MIP:

\[
\begin{align*}
\text{maximize} & \quad (1, 2, 3, 4, 5, 0, \frac{3}{2}, 3 - \frac{1}{2a}) \cdot x \\
\text{subject to} & \quad \begin{pmatrix} 2 & 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \end{pmatrix} x = \begin{pmatrix} 5 \\ 3 \end{pmatrix} \\
x & \in \{0, 1\}^8
\end{align*}
\]

The solution to the LP relaxation has \( (x_1, \ldots, x_5) = (0, 0, \frac{1}{2}, 1, 1) \) and \( (x_6, x_7, x_8) = \left( 0, \frac{1}{2}, 1 \right) \).

Suppose we branch on \( x_3 \). If we set \( x_3 = 0 \),

The new LP relaxation’s solution has \( (x_1, \ldots, x_5) = \left( 0, \frac{1}{2}, 0, 1, 1 \right) \) and \( (x_6, x_7, x_8) = \left( 0, \frac{1}{2}, 1 \right) \).
Proof idea (continued):
For an example’s sake, suppose \( n = 8 \). Consider the MIP:

\[
\begin{align*}
\text{maximize} & \quad \left(1, 2, 3, 4, 5, 0, \frac{3}{2}, 3 - \frac{1}{2a}\right) \cdot x \\
\text{subject to} & \quad \begin{pmatrix} 2 & 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \end{pmatrix} x = \begin{pmatrix} 5 \\ 3 \end{pmatrix} \\
x & \in \{0, 1\}^8
\end{align*}
\]

The solution to the LP relaxation has \( (x_1, \ldots, x_5) = \left(0, 0, \frac{1}{2}, 1, 1\right) \) and \( (x_6, x_7, x_8) = \left(0, \frac{1}{2}, 1\right) \).

Suppose we branch on \( x_3 \). If we set \( x_3 = 0 \),

The new LP relaxation’s solution has \( (x_1, \ldots, x_5) = \left(0, \frac{1}{2}, 0, 1, 1\right) \) and \( (x_6, x_7, x_8) = \left(0, \frac{1}{2}, 1\right) \).
Proof idea (continued):
For an example’s sake, suppose \( n = 8 \). Consider the MIP:

\[
\begin{align*}
\text{maximize} & \quad (1, 2, 3, 4, 5, 0, \frac{3}{2}, 3 - \frac{1}{2a}) \cdot x \\
\text{subject to} & \quad \begin{pmatrix} 2 & 2 & 2 & 2 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 \end{pmatrix} x = \begin{pmatrix} 5 \\ 3 \end{pmatrix} \\
x & \in \{0, 1\}^8
\end{align*}
\]

The solution to the LP relaxation has \((x_1, \ldots, x_5) = \left(0, 0, \frac{1}{2}, 1, 1\right)\) and \((x_6, x_7, x_8) = \left(0, \frac{1}{2}, 1\right)\).

Suppose we branch on \(x_3\). If we set \(x_3 = 1\),

The new LP relaxation’s solution has \((x_1, \ldots, x_5) = \left(0, 0, 1, \frac{1}{2}, 1\right)\) and \((x_6, x_7, x_8) = \left(0, \frac{1}{2}, 1\right)\).
Proof idea (continued):
For an example’s sake, suppose $n = 8$. Consider the MIP:

\[
\begin{align*}
\text{maximize} & \quad (1, 2, 3, 4, 5, 0, \frac{3}{2}, 3 - \frac{1}{2a}) \cdot x \\
\text{subject to} & \quad \begin{pmatrix} 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 \end{pmatrix} x = \begin{pmatrix} 5 \\ 3 \end{pmatrix} \\
x & \in \{0, 1\}^8
\end{align*}
\]

The solution to the LP relaxation has $(x_1, \ldots, x_5) = (0, 0, \frac{1}{2}, 1, 1)$ and $(x_6, x_7, x_8) = (0, \frac{1}{2}, 1)$.

Suppose we branch on $x_7$. If we set $x_7 = 0$,

The new LP relaxation’s solution has $(x_1, \ldots, x_5) = (0, 0, \frac{1}{2}, 1, 1)$ and $(x_6, x_7, x_8) = (\frac{1}{2}, 0, 1)$. 
Proof idea (continued):
For an example’s sake, suppose $n = 8$. Consider the MIP:

$$\begin{align*}
\text{maximize} & \quad \left(1, 2, 3, 4, 5, 0, \frac{3}{2}, 3 - \frac{1}{2a} \right) \cdot x \\
\text{subject to} & \quad \begin{pmatrix} 2 & 2 & 2 & 2 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \end{pmatrix} x = \begin{pmatrix} 5 \\ 3 \end{pmatrix} \\
x & \in \{0, 1\}^8
\end{align*}$$

The solution to the LP relaxation has $(x_1, \ldots, x_5) = \left(0, 0, \frac{1}{2}, 1, 1 \right)$ and $(x_6, x_7, x_8) = \left(0, \frac{1}{2}, 1 \right)$.

Suppose we branch on $x_7$. If we set $x_7 = 1$,

The new LP relaxation’s solution has $(x_1, \ldots, x_5) = \left(0, 0, \frac{1}{2}, 1, 1 \right)$ and $(x_6, x_7, x_8) = \left(0, 1, \frac{1}{2}, 1 \right)$. 

Worst-Case Distributions for Discretization

Proof idea (continued):
For an example’s sake, suppose $n = 8$. Consider the MIP:

\[
\begin{align*}
\text{maximize} & \quad \left(1, 2, 3, 4, 5, 0, \frac{3}{2}, 3 - \frac{1}{2a}\right) \cdot x \\
\text{subject to} & \quad \left(2 \ 2 \ 2 \ 2 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 2 \ 2 \ 2 \right) x = \left(5 \ 3 \right) \\
x & \in \{0, 1\}^8
\end{align*}
\]

By this reasoning, we can calculate that

\[
\begin{align*}
\mu \cdot \text{score}_1(Q, 3) + (1 - \mu)\text{score}_2(Q, 3) &= \frac{1}{2} \\
\mu \cdot \text{score}_1(Q, 7) + (1 - \mu)\text{score}_2(Q, 7) &= \frac{3}{4} - \frac{\mu}{4a}
\end{align*}
\]

\[
\text{score}_1(Q, i) = \min \left\{ c_Q - c_{Q_i}^-, \ c_Q - c_{Q_i}^+ \right\} \quad \text{score}_2(Q, i) = \max \left\{ c_Q - c_{Q_i}^-, \ c_Q - c_{Q_i}^+ \right\}
\]
Worst-Case Distributions for Discretization

Proof idea (continued):
For an example’s sake, suppose \( n = 8 \). Consider the MIP:

\[
\begin{align*}
\text{maximize} & \quad (1, 2, 3, 4, 5, 0, \frac{3}{2}, 3 - \frac{1}{2a}) \cdot x \\
\text{subject to} & \quad (2 \ 2 \ 2 \ 2 \ 2 \ 0 \ 0 \ 0) x = (5) \\
x & \in \{0, 1\}^8
\end{align*}
\]

By this reasoning, we can calculate that

\[
\mu \cdot \text{score}_1(Q, 3) + (1 - \mu)\text{score}_2(Q, 3) = \frac{1}{2}
\]

\[
\mu \cdot \text{score}_1(Q, 7) + (1 - \mu)\text{score}_2(Q, 7) = \frac{3}{4} - \frac{\mu}{4a}
\]

B&B will branch on \( x_7 \) if \( \mu \leq a \) and otherwise it will branch on \( x_3 \).
Proof idea (continued):

For an example’s sake, suppose \( n = 8 \). Consider the MIP:

maximize \( \left( 1, 2, 3, 4, 5, \frac{3}{2}, 3 - \frac{1}{2a} \right) \cdot x \)

subject to

\[
\begin{pmatrix}
2 & 2 & 2 & 2 & 0 & 0 & 0 & 2 & 2 & 2
\end{pmatrix} x = \begin{pmatrix} 5 \\ 3 \end{pmatrix}
\]

\( x \in \{0, 1\}^8 \)

B&B will branch on \( x_7 \) if \( \mu \leq a \) and otherwise it will branch on \( x_3 \).

Corresponds to the small version of Jeroslow’s instance
Proof idea (continued):
For an example’s sake, suppose $n = 8$. Consider the MIP:

$$\begin{align*}
\text{maximize} & \quad (1, 2, 3, 4, 5, 0, \frac{3}{2}, 3 - \frac{1}{2a}) \cdot x \\
\text{subject to} & \quad \begin{pmatrix} 2 & 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \end{pmatrix} x = \begin{pmatrix} 5 \\ 3 \end{pmatrix} \\
x & \in \{0, 1\}^8
\end{align*}$$

B&B will branch on $x_7$ if $\mu \leq a$ and otherwise it will branch on $x_3$. 

Corresponds to the big version of Jeroslow’s instance
Proof idea (continued):
For an example’s sake, suppose \( n = 8 \). Consider the MIP:

\[
\begin{align*}
\text{maximize} & \quad \left(1, 2, 3, 4, 5, 0, \frac{3}{2}, 3 - \frac{1}{2a}\right) \cdot x \\
\text{subject to} & \quad \begin{pmatrix} 2 & 2 & 2 & 2 & 0 & 0 & 0 \ 0 & 0 & 0 & 0 & 2 & 2 & 2 \end{pmatrix} x = \begin{pmatrix} 5 \\ 3 \end{pmatrix} \\
x \in \{0, 1\}^8
\end{align*}
\]

When \( \mu \leq a \), we continuing tracking B&B’s progress to make sure it only branches on variables from the small instance \( \{x_6, x_7, x_8\} \) before figuring out the MIP is infeasible. The tree will have constant size.
Proof idea (continued):
For an example’s sake, suppose $n = 8$. Consider the MIP:

maximize $\left(1, 2, 3, 4, 5, 0, \frac{3}{2}, 3 - \frac{1}{2a}\right) \cdot x$

subject to

$\begin{bmatrix} 2 & 2 & 2 & 2 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 0 \end{bmatrix} x = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$

$x \in \{0, 1\}^8$

When $\mu > a$, we prove by induction that if B&B has only branched on variables from the big instance ($\{x_1, ..., x_5\}$), it will continue to only branch on those variables.

We also prove it will branch on about half of these variables on each path.

The tree will have exponential size.
Proof idea (continued):
For an example’s sake, suppose $n = 8$. Consider the MIP:

$maximize \quad \left(1, 2, 3, 4, 5, 0, \frac{3}{2}, 3 - \frac{1}{2a}\right) \cdot x$

subject to \[
\begin{pmatrix}
2 & 2 & 2 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 2 & 2 \\
\end{pmatrix} x = \begin{pmatrix} 5 \\
3 \end{pmatrix}
\]

$x \in \{0, 1\}^8$

(We also generalize to $n$ variables, rather than 8.)

The proof for the opposite direction is similar.
Outline

1. Branch-and-Bound
   i. Algorithm Overview
   ii. Variable Selection Policies
2. Learning Algorithms
   i. First-try: Discretization
   ii. Our Approach
3. Generalization Guarantees
4. Experiments
5. Conclusion and Future Directions
A Useful Lemma

**Lemma**

For any 2 scoring rules \(\text{score}_1, \text{score}_2\) and any MIP instance \(Q\), there exist at most \(O((\#\text{ variables})^{(\#\text{ variables})^2})\) intervals partitioning \([0,1]\) such that for any interval \([a, b]\), the tree B&B builds given input \(Q\) using the scoring rule \(\mu \cdot \text{score}_1 + (1 - \mu) \cdot \text{score}_2\) is fixed over all \(\mu \in [a, b]\).
For any two scoring rules $\text{score}_1$, $\text{score}_2$ and any MIP instance $Q$, there exist at most $O((\# \text{ variables})^{\# \text{ variables}^2})$ intervals partitioning $[0,1]$ such that for any interval $[a, b]$, the tree B&B builds given input $Q$ using the scoring rule $\mu \cdot \text{score}_1 + (1 - \mu) \cdot \text{score}_2$ is fixed over all $\mu \in [a, b]$.

**A Useful Lemma**

*Much* smaller in our experiments!

Can be generalized to $d$-dimensional parameters.
A Useful Lemma

Lemma

For any 2 scoring rules $\text{score}_1, \text{score}_2$ and any MIP instance $Q$, there exist at most $O((\# \text{ variables})^2)$ intervals partitioning $[0,1]$ such that for any interval $[a, b]$, the tree B&B builds given input $Q$ using the scoring rule $\mu \cdot \text{score}_1 + (1 - \mu) \cdot \text{score}_2$ is fixed over all $\mu \in [a, b]$.

Proof sketch.

\[\mu \cdot \text{score}_1(Q, 1) + (1 - \mu) \cdot \text{score}_2(Q, 1)\]

\[\mu \cdot \text{score}_1(Q, 3) + (1 - \mu) \cdot \text{score}_2(Q, 3)\]

\[\mu \cdot \text{score}_1(Q, 2) + (1 - \mu) \cdot \text{score}_2(Q, 2)\]
A Useful Lemma

Lemma

For any 2 scoring rules score₁, score₂ and any MIP instance Q, there exist at most \(O\left(\frac{\# \text{variables} \times (\# \text{variables})^2}{\# \text{variables}}\right)\) intervals partitioning [0,1] such that for any interval \([a, b]\), the tree B&B builds given input Q using the scoring rule \(\mu \cdot \text{score}_1 + (1 - \mu) \cdot \text{score}_2\) is fixed over all \(\mu \in [a, b]\).

Proof sketch.

Using any \(\mu'\) from the yellow interval results in this subtree:

\[ x_2 = 0 \quad x_2 = 1 \]
A Useful Lemma

**Lemma**

For any 2 scoring rules $\text{score}_1$, $\text{score}_2$ and any MIP instance $Q$, there exist at most $O((\# \text{ variables})^{(\# \text{ variables})^2})$ intervals partitioning $[0,1]$ such that for any interval $[a, b]$, the tree B&B builds given input $Q$ using the scoring rule $\mu \cdot \text{score}_1 + (1 - \mu) \cdot \text{score}_2$ is fixed over all $\mu \in [a, b]$.

**Proof sketch.**

Using any $\mu'$ from the yellow interval results in this subtree:

$x_2 = 0$

$x_2 = 1$

$x_2$ then $x_3$ branched on

$x_2$ then $x_1$ branched on

$\mu \cdot \text{score}_1(Q_2^-, 1) + (1 - \mu) \cdot \text{score}_2(Q_2^-, 1)$

$\mu \cdot \text{score}_1(Q_2^-, 3) + (1 - \mu) \cdot \text{score}_2(Q_2^-, 3)$
A Useful Lemma

Lemma

For any 2 scoring rules $\text{score}_1, \text{score}_2$ and any MIP instance $Q$, there exist at most $O((\# \text{ variables})^2)$ intervals partitioning $[0,1]$ such that for any interval $[a, b]$, the tree B&B builds given input $Q$ using the scoring rule $\mu \cdot \text{score}_1 + (1 - \mu) \cdot \text{score}_2$ is fixed over all $\mu \in [a, b]$.

Proof sketch.

Using any $\mu'$ from the yellow-blue interval results in this subtree:

$x_3 = 0$

$x_3 = 1$
For any 2 scoring rules $\text{score}_1, \text{score}_2$ and any MIP instance $Q$, there exist at most $O((\# \text{variables})^2)$ intervals partitioning $[0,1]$ such that for any interval $[a, b]$, the tree B&B builds given input $Q$ using the scoring rule $\mu \cdot \text{score}_1 + (1 - \mu) \cdot \text{score}_2$ is fixed over all $\mu \in [a, b]$.

Proof sketch.

We can continue subdividing the real line into intervals over which the variable selection order is fixed.

There are only a finite number of variables, so we can only subdivide a finite number of times.

Formally, the proof follows by induction on the tree depth.
**Algorithm**

**Input:** A set of MIP instances; two scoring rules \( \text{score}_1, \text{score}_2 \)

For each MIP instance, set \( \mu = 0 \). While \( \mu < 1 \):

1. Run B&B using \( \mu \cdot \text{score}_1 + (1 - \mu) \cdot \text{score}_2 \), resulting in a tree \( \mathcal{T} \)
2. Find the interval \([\mu, \mu']\) where if B&B is run using the scoring rule
   \[ \mu'' \cdot \text{score}_1 + (1 - \mu'') \cdot \text{score}_2 \]
   for any \( \mu'' \in [\mu, \mu'] \), the resulting tree will be \( \mathcal{T} \) (takes a little bookkeeping).
3. Set \( \mu = \mu' \)

**Return:** Any \( \mu \) from the interval minimizing average cost.
Outline

1. Branch-and-Bound
   i. Algorithm Overview
   ii. Variable Selection Policies
2. Learning Algorithms
   i. First-try: Discretization
   ii. Our Approach
3. Generalization Guarantees
4. Experiments
5. Conclusion and Future Directions
The parameter $\hat{\mu}$ is optimal over the set of samples, but is it also (nearly) optimal in expectation over the distribution $\mathcal{D}$?
Generalization

Application-Specific Distribution $\mathcal{D}$

$\{A^{(1)}, b^{(1)}, c^{(1)}\}, \ldots, \{A^{(m)}, b^{(m)}, c^{(m)}\}$

Algorithm Designer

E.g., search tree size.

$\hat{\mu} = \arg\min_{\mu} \{\sum_{i=1}^{m} \text{cost}_{\mu}(A^{(i)}, b^{(i)}, c^{(i)})\}.$
Generalization

Application-Specific Distribution $\mathcal{D}$

$\{A^{(1)}, b^{(1)}, c^{(1)}\}, \ldots, \{A^{(m)}, b^{(m)}, c^{(m)}\}$

$\hat{\mu} = \arg\min_{\mu} \{\sum_{i=1}^{m} \text{cost}_{\mu}(A^{(i)}, b^{(i)}, c^{(i)})\}$.

Does that mean $\mathbb{E}_{(A,b,c) \sim \mathcal{D}}[\text{cost}_{\hat{\mu}}(A, b, c)] - \min_{\mu}\{\mathbb{E}_{(A,b,c) \sim \mathcal{D}}[\text{cost}_{\mu}(A, b, c)]\} < \varepsilon$?
The **more samples** the algorithm designer has, the more confident he can be that **low average cost** over the samples implies **low empirical cost** on the distribution.

How many samples are enough?
Suppose $\mathcal{C} = \{\text{cost}_\mu : \mu \in [0,1]\}$, and let $[0, U]$ be the range of cost $\mu$ for all $\mu$. Given a set of samples $S = \{(A^{(1)}, b^{(1)}, c^{(1)}), ..., (A^{(m)}, b^{(m)}, c^{(m)})\}$, let

$$\hat{\mu} = \arg\min_{\mu} \left\{ \sum_{i=1}^{m} \text{cost}_\mu(A^{(i)}, b^{(i)}, c^{(i)}) \right\}.$$

For all $\epsilon, \delta > 0$, $m = \tilde{O} \left( \left( \frac{U}{\epsilon} \right)^2 \text{Pdim}(\mathcal{C}) \right)$ samples are sufficient to ensure that with high probability over the draw of $S \sim \mathcal{D}^m$,

$$\mathbb{E}_{(A, b, c) \sim \mathcal{D}} [\text{cost}_{\hat{\mu}}(A, b, c)] - \min_{\mu} \{\mathbb{E}_{(A, b, c) \sim \mathcal{D}} [\text{cost}_\mu(A, b, c)]\} < \epsilon.$$
What is Pseudo-Dimension?

**Pseudo-dimension** is essentially **VC dimension** for real-valued functions. It measures the **intrinsic complexity** of a function class.

\[ \mu \in [0,1] \]
Pseudo-Dimension for MIPs

**Theorem**

If, for any MIP \((A, b, c)\), \(\text{cost}_{(A,b,c)}(\mu)\) is piecewise constant with at most \(k\) pieces, then \(\text{Pdim}(\{\text{cost}_{\mu}: \mu \in [0,1]\}) = O(\log k)\).
Recall: A Useful Lemma

**Lemma**

For any 2 scoring rules \( \text{score}_1, \text{score}_2 \) and any MIP instance \( Q \), there exist at most \( O\left((\# \text{variables})^{\# \text{variables}^2}\right) \) intervals partitioning \([0,1]\) such that for any interval \([a, b]\), the tree B&B builds given input \( Q \) using the scoring rule \( \mu \cdot \text{score}_1 + (1 - \mu) \cdot \text{score}_2 \) is fixed over all \( \mu \in [a, b] \).
Recall: A Useful Lemma

<table>
<thead>
<tr>
<th>Lemma</th>
</tr>
</thead>
<tbody>
<tr>
<td>For any 2 scoring rules ( \text{score}_1, \text{score}_2 ) and any MIP instance ( Q ), there exist at most ( O \left( \left( # \text{variables} \right)^2 \right) ) intervals partitioning ([0,1]) such that for any interval ([a, b]), the tree B&amp;B builds given input ( Q ) using the scoring rule ( \mu \cdot \text{score}_1 + (1 - \mu) \cdot \text{score}_2 ) is fixed over all ( \mu \in [a, b] ).</td>
</tr>
</tbody>
</table>

For any MIP \((A, b, c)\), \( \text{cost}_{(A,b,c)}(\mu) \) is piecewise constant with at most \( O \left( \left( \# \text{variables} \right)^2 \right) \) pieces.

\[ \text{Pdim} \left( \{ \text{cost}_\mu : \mu \in [0,1] \} \right) = O \left( \left( \# \text{variables} \right)^2 \log(\# \text{variables}) \right). \]
Generalization Guarantees for MIPs

**Theorem**

Suppose $\mathcal{C} = \{\text{cost}_\mu: \mu \in [0,1]\}$, and let $[0, U]$ be the range of $\text{cost}_\mu$ for all $\mu$. Given a set of samples $S = \{(A^{(1)}, b^{(1)}, c^{(1)}), \ldots, (A^{(m)}, b^{(m)}, c^{(m)})\}$, let

$$\hat{\mu} = \arg\min_{\mu} \left\{ \sum_{i=1}^{m} \text{cost}_\mu(A^{(i)}, b^{(i)}, c^{(i)}) \right\}.$$

For all $\epsilon, \delta > 0, m = \tilde{O} \left( \left( \frac{U(\# \text{ variables})}{\epsilon} \right)^2 \right)$ samples are sufficient to ensure that with high probability over the draw of $S \sim \mathcal{D}^m$,

$$\mathbb{E}_{(A,b,c) \sim \mathcal{D}} \left[ \text{cost}_{\hat{\mu}}(A, b, c) \right] - \min_{\mu} \left\{ \mathbb{E}_{(A,b,c) \sim \mathcal{D}} \left[ \text{cost}_\mu(A, b, c) \right] \right\} < \epsilon.$$
Outline

1. Branch-and-Bound
   i. Algorithm Overview
   ii. Variable Selection Policies
2. Learning Algorithms
   i. First-try: Discretization
   ii. Our Approach
3. Generalization Guarantees
4. Experiments
5. Conclusion and Future Directions
Experiments: Tuning the Linear Rule

Let:

\[
\begin{align*}
\text{score}_1(Q, i) &= \min \{c_Q - c_{Q_i^-}, c_Q - c_{Q_i^+}\} \\
\text{score}_2(Q, i) &= \max \{c_Q - c_{Q_i^-}, c_Q - c_{Q_i^+}\}
\end{align*}
\]

Branch on the fractional variable \(x_i\) that maximizes:

\[
\text{score}(Q, i) = \mu \cdot \text{score}_1(Q, i) + (1 - \mu) \cdot \text{score}_2(Q, i)
\]

This is the linear rule [Linderoth & Savelsbergh, 1999]

Our parameterized rule

Branch on the fractional variable \(x_i\) that maximizes:

\[
\text{score}(Q, i) = \mu \cdot \text{score}_1(Q, i) + (1 - \mu) \cdot \text{score}_2(Q, i)
\]
Experiments: Combinatorial Auctions

Additional Experiments

**Clustering:**
5 clusters, 35 nodes, 500 instances

**Facility location:**
70 facilities, 70 customers, 500 instances

**Agnostically learning linear separators:**
50 points in $\mathbb{R}^2$, 500 instances
Outline

1. Branch-and-Bound
   i. Algorithm Overview
   ii. Variable Selection Policies
2. Learning Algorithms
   i. First-try: Discretization
   ii. Our Approach
3. Generalization Guarantees
4. Experiments
5. Conclusion and Future Directions
Conclusion

- We study B&B, one of the most widely-used algorithms for nonconvex problems.
- We show how to use machine learning to determine a nearly optimal weighting of any set of variable selection scoring rules for the application at hand.
- Empirically, we show learning an optimal weighting can dramatically reduce tree size.
  - We prove that this improvement can even be exponential.
- We provide the first sample complexity bounds for tree search algorithm selection.
- Our theory also applies to other tree search algorithms, e.g., for solving CSPs.
Future Directions

• How can we train faster?
  – We don’t want to compute every tree B&B will make for every instance in the training set
  – Can we use a best-arm-identification approach?
  – Is it possible to train on small MIP instances and then apply the learned policies on large MIP instances?

• What other tree-building applications can we apply our techniques to?
  – E.g., building decision trees and taxonomies

• How can we attack other learning problems in MIP?
  – E.g., node-selection policies
Questions?