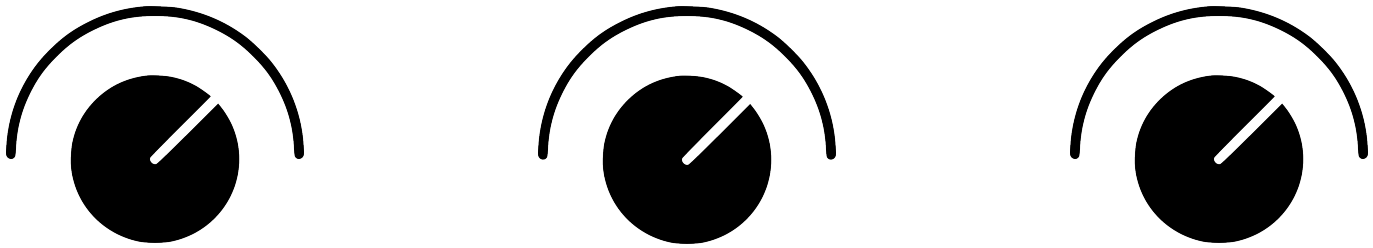
Refined bounds for algorithm configuration:
The knife-edge of dual class approximability

Nina Balcan, Tuomas Sandholm, Ellen Vitercik

Algorithms typically come with **many tunable parameters**

Significant impact on runtime, solution quality, …

Hand-tuning is **time-consuming**, **tedious**, and **error-prone**

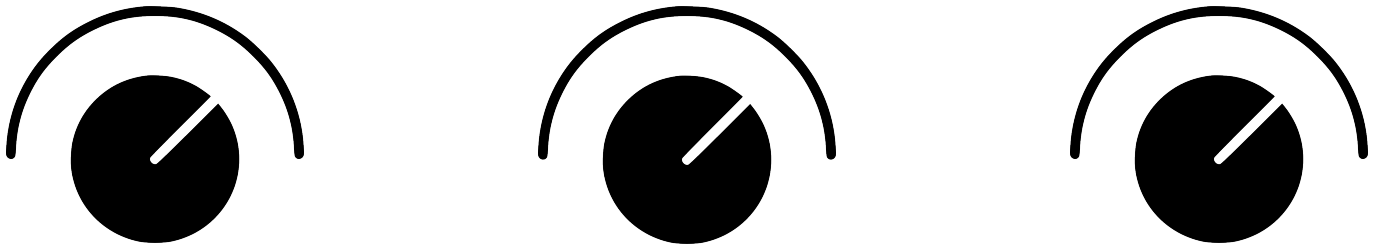


Automated algorithm configuration

Goal: Automate algorithm configuration via machine learning

Algorithmically find good parameter settings
using a **set of "typical" inputs** from application at hand

Training set



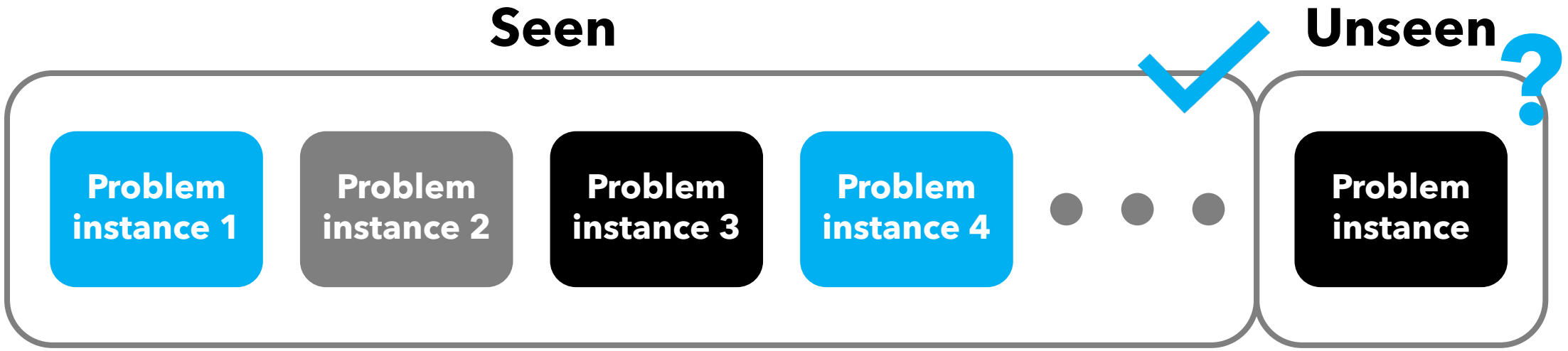
Automated configuration procedure

1. Fix parameterized algorithm (e.g., CPLEX)
2. Receive set $\mathcal{S}$ of "typical" inputs from unknown distribution

3. Return parameter setting with good avg performance over $\mathcal{S}$

Runtime, solution quality, memory usage, etc.

Automated configuration procedure

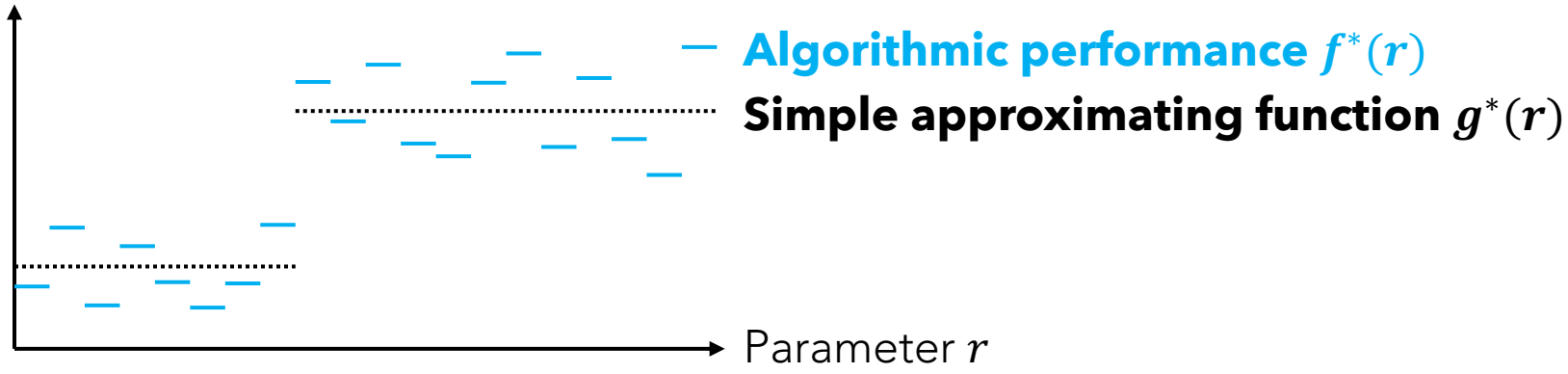


Key question (focus of talk):
Will those parameters have good **expected** performance?

Overview of main result

Key question (focus of talk):
Will those parameters have good **expected** performance?

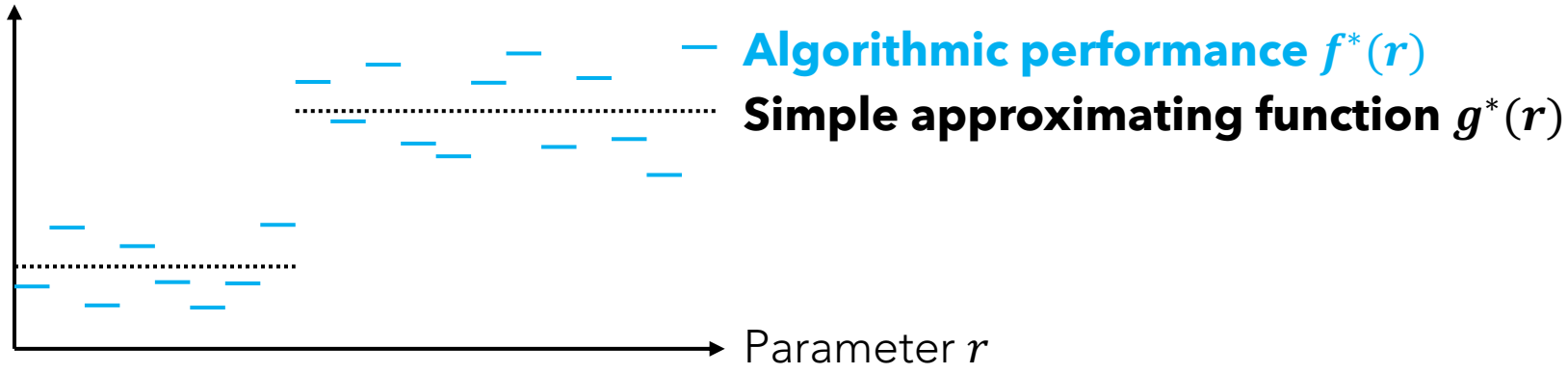
"Yes" when algorithmic performance as function of parameters can be approximated by a simple function



Overview of main result

Observe this structure, e.g.,

in integer programming algorithm configuration

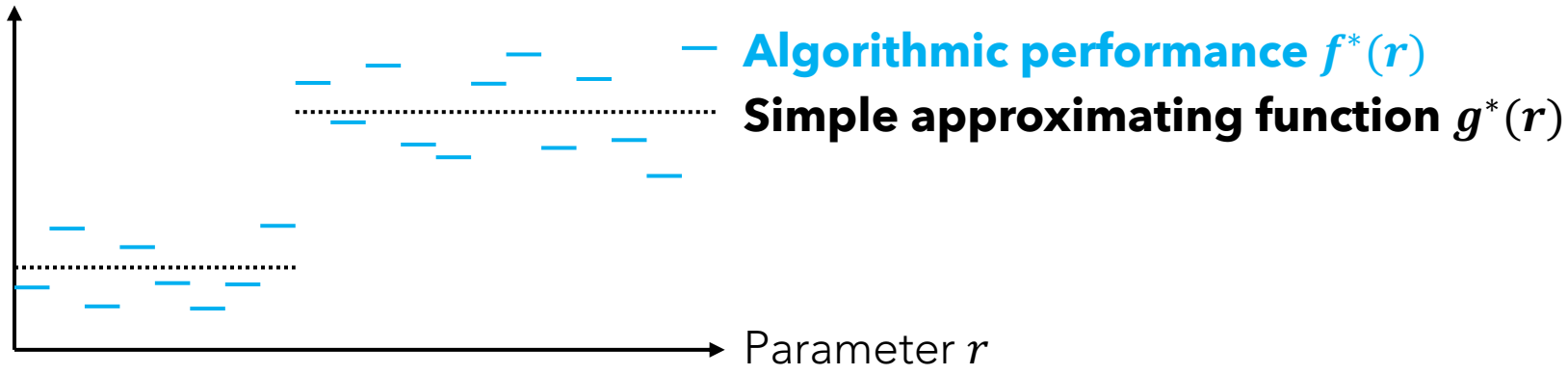


Overview of main result: a dichotomy

If approximation holds under the L^∞-norm:

We provide strong guarantees

$\sup_r |f^*(r) - g^*(r)|$ is small



Overview of main result: a dichotomy

If approximation holds under the L^∞-norm:

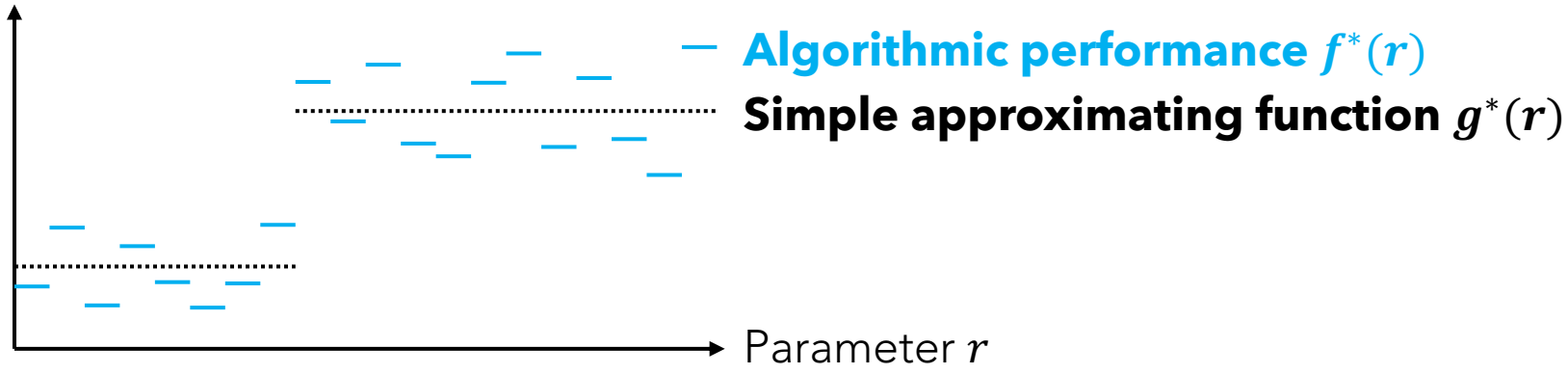
We provide strong guarantees

If approximation only holds under the L^p-norm for $p < \infty$:

Not possible to provide strong guarantees in worst case

$$\sqrt[p]{\int |f^*(r) - g^*(r)|^p \, dr}$$

is small



Model

Model

$\mathcal{X}$: Set of all inputs (e.g., integer programs)

$\mathbb{R}^d$: Set of all parameter settings (e.g., CPLEX parameters)

Standard assumption: Unknown distribution $\mathcal{D}$ over inputs

E.g., represents scheduling problem airline solves day-to-day

"Algorithmic performance"

$f_{\boldsymbol{r}}(x)$ = utility of algorithm parameterized by $\boldsymbol{r} \in \mathbb{R}^d$ on input x

E.g., runtime, solution quality, memory usage, …

Assume $f_{\boldsymbol{r}}(x) \in [-1,1]$

Can be generalized to $f_{\boldsymbol{r}}(x) \in [-H, H]$

Generalization bounds

Generalization bounds

Key question: For any parameter setting $\boldsymbol{r}$,
Does good **avg** utility on training set imply good **exp** utility?

Formally: Given samples $x_1, \ldots, x_N \sim \mathcal{D}$, for any $\boldsymbol{r}$,

$$\left| \frac{1}{N} \sum_{i=1}^{N} f_{\boldsymbol{r}}(x_i) - \mathbb{E}_{x \sim \mathcal{D}}[f_{\boldsymbol{r}}(x)] \right| \leq \textbf{?}$$

Empirical average utility

Generalization bounds

Key question: For any parameter setting $\boldsymbol{r}$,
Does good **avg** utility on training set imply good **exp** utility?

Formally: Given samples $x_1, \ldots, x_N \sim \mathcal{D}$, for any $\boldsymbol{r}$,

$$\left| \frac{1}{N} \sum_{i=1}^{N} f_{\boldsymbol{r}}(x_i) - \underbrace{\mathbb{E}_{x \sim \mathcal{D}}[f_{\boldsymbol{r}}(x)]}_{\textbf{Expected utility}} \right| \leq \textbf{?}$$

Generalization bounds

Key question: For any parameter setting $\boldsymbol{r}$,
Does good **avg** utility on training set imply good **exp** utility?

Formally: Given samples $x_1, \ldots, x_N \sim \mathcal{D}$, for any $\boldsymbol{r}$,

$$\left| \frac{1}{N} \sum_{i=1}^{N} f_{\boldsymbol{r}}(x_i) - \mathbb{E}_{x \sim \mathcal{D}}[f_{\boldsymbol{r}}(x)] \right| \leq \textbf{?}$$

Typically, answer by bounding the **intrinsic complexity** of

$$\mathcal{F} = \{ f_{\boldsymbol{r}} \mid \boldsymbol{r} \in \mathbb{R}^d \}$$

Generalization bounds

Challenge: Class $\mathcal{F} = \{f_{\boldsymbol{r}} : \mathcal{X} \to \mathbb{R} \mid \boldsymbol{r} \in \mathbb{R}^d\}$ is gnarly

E.g., in integer programming algorithm configuration:

- Each domain element is an IP
- Unclear how to plot or visualize functions $f_{\boldsymbol{r}}$
- No obvious notions of Lipschitzness or smoothness to rely on

Dual functions

Dual classes

$f_{\boldsymbol{r}}(x) =$ utility of algorithm parameterized by $\boldsymbol{r} \in \mathbb{R}^d$ on input x

$\mathcal{F} = \{f_{\boldsymbol{r}} : \mathcal{X} \to \mathbb{R} \mid \boldsymbol{r} \in \mathbb{R}^d\}$ **"Primal" function class**

$f_x^*(\boldsymbol{r}) =$ utility as function of parameters

$f_x^*(\boldsymbol{r}) = f_{\boldsymbol{r}}(x)$

$\mathcal{F}^* = \{f_x^* : \mathbb{R}^d \to \mathbb{R} \mid x \in \mathcal{X}\}$ **"Dual" function class**

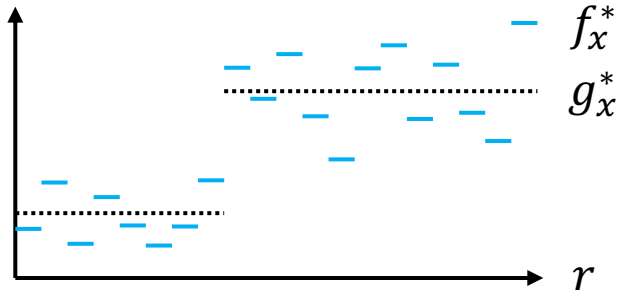
- Dual functions have simple, Euclidean domain
- Often have ample structure can use to bound complexity of $\mathcal{F}$

Dual function approximability

$\mathcal{F} = \{f_{\boldsymbol{r}} \mid \boldsymbol{r} \in \mathbb{R}^d\}$
$\mathcal{G} = \{g_{\boldsymbol{r}} \mid \boldsymbol{r} \in \mathbb{R}^d\}$
} Sets of functions mapping $\mathcal{X}$ to $\mathbb{R}$

Dual class $\mathcal{G}^*$ **(γ, p)-approximates** $\mathcal{F}^*$ if for all $x \in \mathcal{X}$,

$$\|f_x^* - g_x^*\|_p = \sqrt[p]{\int_{\mathbb{R}^d} |f_x^*(\boldsymbol{r}) - g_x^*(\boldsymbol{r})|^p \, d\boldsymbol{r}} \leq \gamma.$$



Main result: Upper bound

Generalization upper bound

$$\left.\begin{aligned}\mathcal{F} &= \{f_{\boldsymbol{r}} \mid \boldsymbol{r} \in \mathbb{R}^d\} \\ \mathcal{G} &= \{g_{\boldsymbol{r}} \mid \boldsymbol{r} \in \mathbb{R}^d\}\end{aligned}\right\} \text{ Sets of functions mapping } \mathcal{X} \text{ to } \mathbb{R}$$

With high probability over the draw of $\mathcal{S} \sim \mathcal{D}^N$, for any $\boldsymbol{r}$,

$$\left|\frac{1}{N}\sum_{x \in \mathcal{S}} f_{\boldsymbol{r}}(x) - \mathop{\mathbb{E}}_{x \sim \mathcal{D}}[f_{\boldsymbol{r}}(x)]\right| = \tilde{O}\left(\frac{1}{N}\sum_{x \in \mathcal{S}} \|f_x^* - g_x^*\|_\infty + \widehat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{G}) + \sqrt{\frac{1}{N}}\right)$$

Average utility over the training set

Generalization upper bound

$$\left.\begin{aligned}\mathcal{F} &= \{f_{\boldsymbol{r}} \mid \boldsymbol{r} \in \mathbb{R}^d\} \\ \mathcal{G} &= \{g_{\boldsymbol{r}} \mid \boldsymbol{r} \in \mathbb{R}^d\}\end{aligned}\right\}$$ Sets of functions mapping $\mathcal{X}$ to $\mathbb{R}$

With high probability over the draw of $\mathcal{S} \sim \mathcal{D}^N$, for any $\boldsymbol{r}$,

$$\left|\frac{1}{N}\sum_{x \in \mathcal{S}} f_{\boldsymbol{r}}(x) - \underbrace{\mathop{\mathbb{E}}_{x \sim \mathcal{D}}[f_{\boldsymbol{r}}(x)]}_{\textbf{Expected utility}}\right| = \tilde{O}\left(\frac{1}{N}\sum_{x \in \mathcal{S}} \|f_x^* - g_x^*\|_\infty + \widehat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{G}) + \sqrt{\frac{1}{N}}\right)$$

Generalization upper bound

$$\left.\begin{aligned}\mathcal{F} &= \{f_{\boldsymbol{r}} \mid \boldsymbol{r} \in \mathbb{R}^d\} \\ \mathcal{G} &= \{g_{\boldsymbol{r}} \mid \boldsymbol{r} \in \mathbb{R}^d\}\end{aligned}\right\}$$ Sets of functions mapping $\mathcal{X}$ to $\mathbb{R}$

With high probability over the draw of $\mathcal{S} \sim \mathcal{D}^N$, for any $\boldsymbol{r}$,

$$\left|\frac{1}{N}\sum_{x \in \mathcal{S}} f_{\boldsymbol{r}}(x) - \mathop{\mathbb{E}}_{x \sim \mathcal{D}}[f_{\boldsymbol{r}}(x)]\right| = \tilde{O}\left(\frac{1}{N}\sum_{x \in \mathcal{S}} \|f_x^* - g_x^*\|_\infty + \widehat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{G}) + \sqrt{\frac{1}{N}}\right)$$

If $\mathcal{G}$ not too complex and $\mathcal{G}^*$ (γ, ∞)-approximates $\mathcal{F}^*$,

Bound approaches $O(\gamma)$ as $N \to \infty$.

Main result: Lower bound

Lower bound

For any γ and $p < \infty$, there exist function classes $\mathcal{F}, \mathcal{G}$ such that:

- Dual class $\mathcal{G}^*$ (γ, p)-approximates $\mathcal{F}^*$
- $\mathcal{G}$ is **very simple** Rademacher complexity is 0
- $\mathcal{F}$ is **very complex** Rademacher complexity is $\frac{1}{2}$
  - **Not possible** to provide generalization bounds in worst case

Experiments

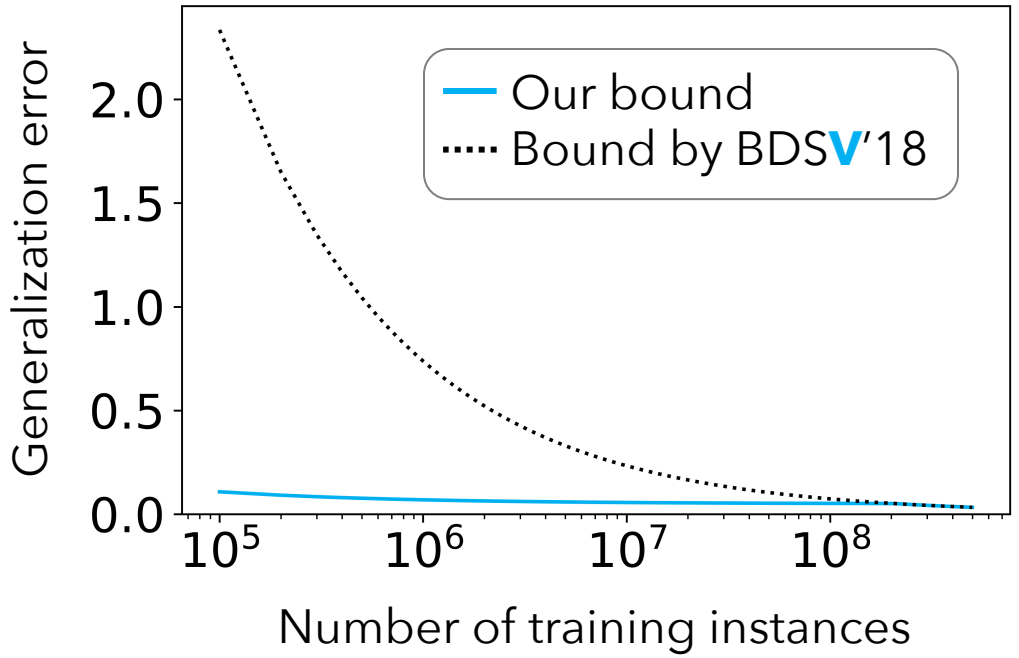
Experiments: Integer programming

Tune integer programming solver parameters

Also studied by Balcan, Dick, Sandholm, **Vitercik** [ICML'18]

Distributions over auction IPs

[Leyton-Brown, Pearson, Shoham, EC'00]



Conclusion

Conclusion

- Provided generalization bounds for algorithm configuration
- Apply whenever utility as function of parameters is **"approximately simple"**
- Connection between learnability and approximability is **balanced on a knife-edge**
  - If approximation holds under L^∞-norm, can provide strong bounds
  - If holds under L^p-norm for $p < \infty$, not possible to provide bounds
- Experiments demonstrate strength of these bounds