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

Nina Balcan, Tuomas Sandholm, Ellen Vitercik

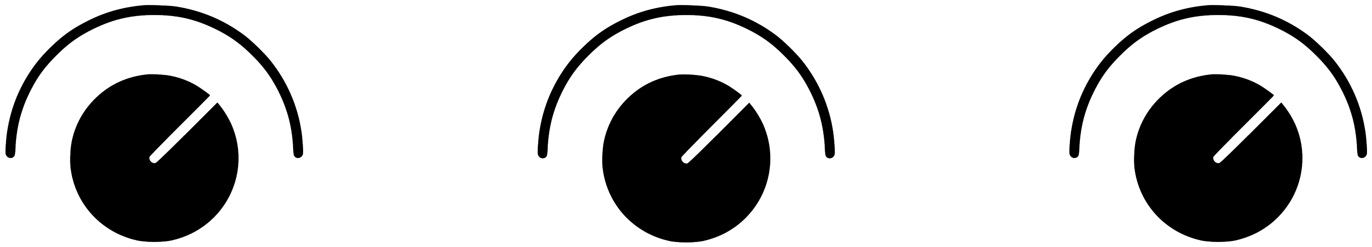
Carnegie Mellon University

INFORMS Annual Meeting 2020

Algorithms typically come with **many tunable parameters**

- Significant impact on runtime, solution quality, …

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

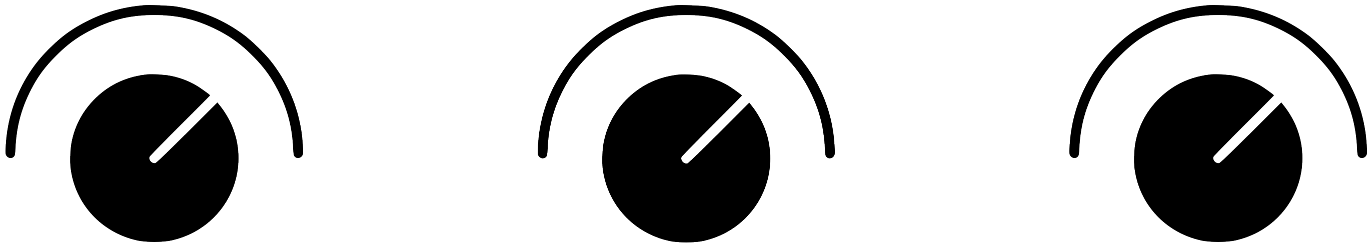


Automated algorithm configuration

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

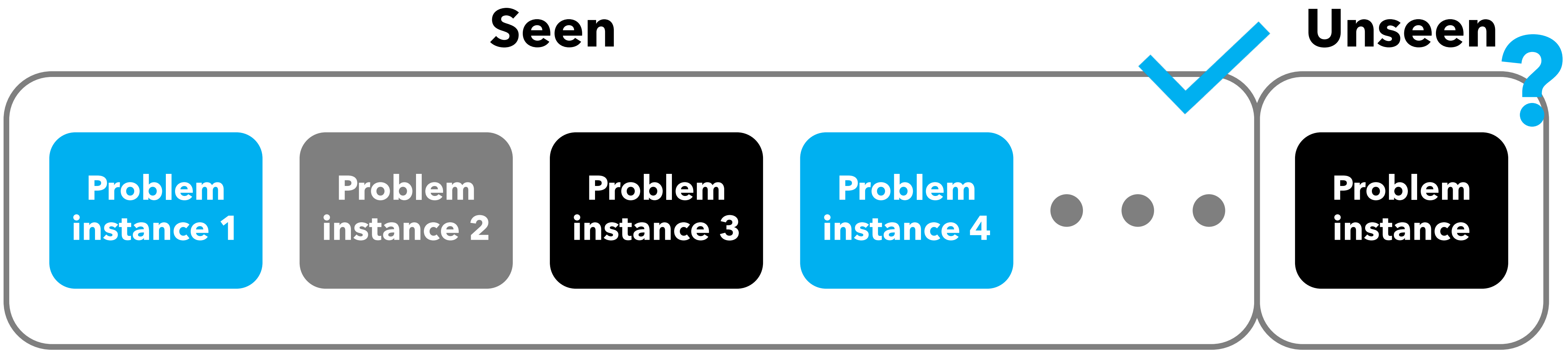
1. Fix parameterized algorithm (e.g., CPLEX)
2. Receive set $\mathcal{S}$ of "typical" inputs from unknown distribution $\mathcal{D}$

Problem instance 1 | Problem instance 2 | Problem instance 3 | Problem instance 4 | • • •

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?

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

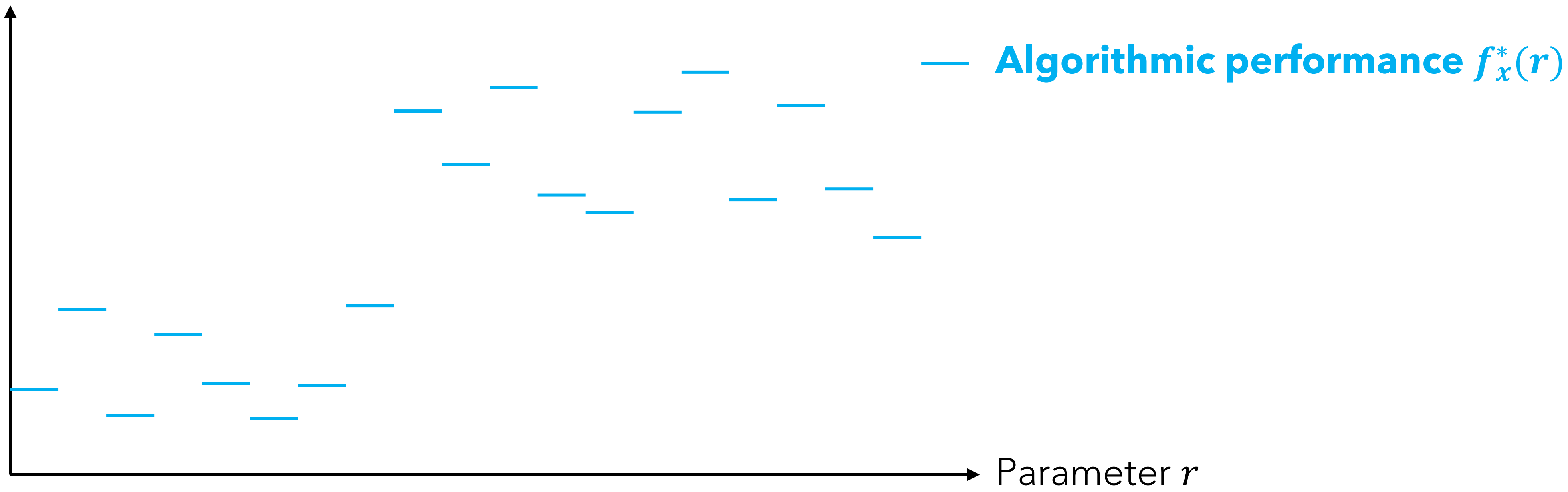
Can be generalized to $f_{\boldsymbol{r}}(x) \in [-H, H]$

Dual functions

Dual functions

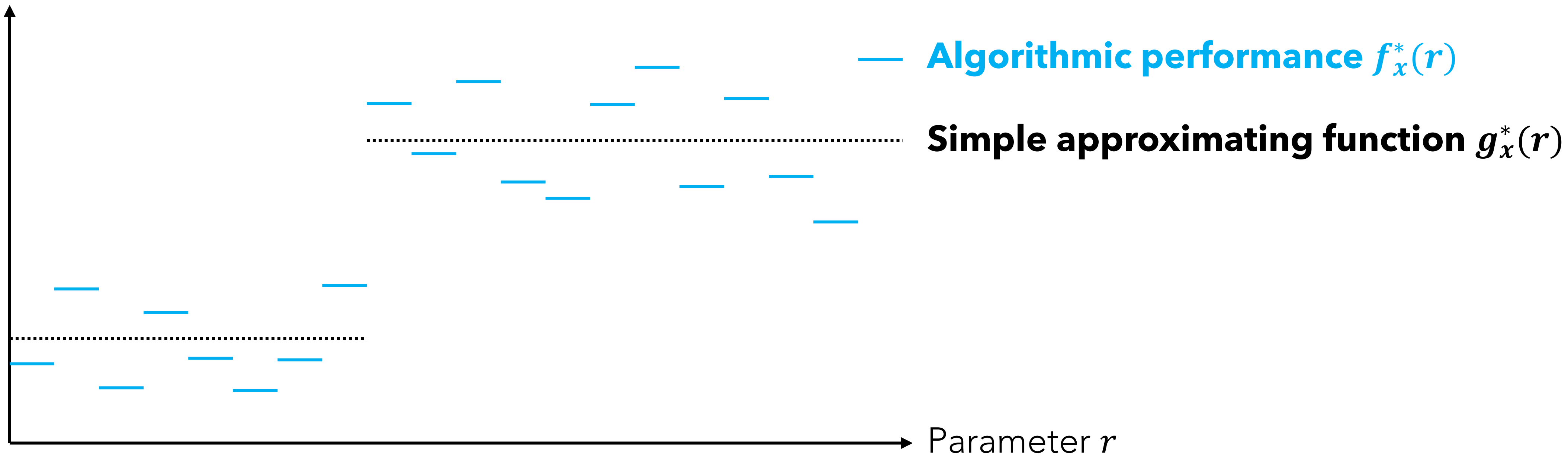
For input x, "dual function" f_x^* measures

algorithmic performance as a function of the parameters



Dual functions

Duals are often well-approximated by a "simple function"
e.g., in integer programming algorithm configuration



Main result

Generalization bounds

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

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

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

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 $\boldsymbol{r}$,

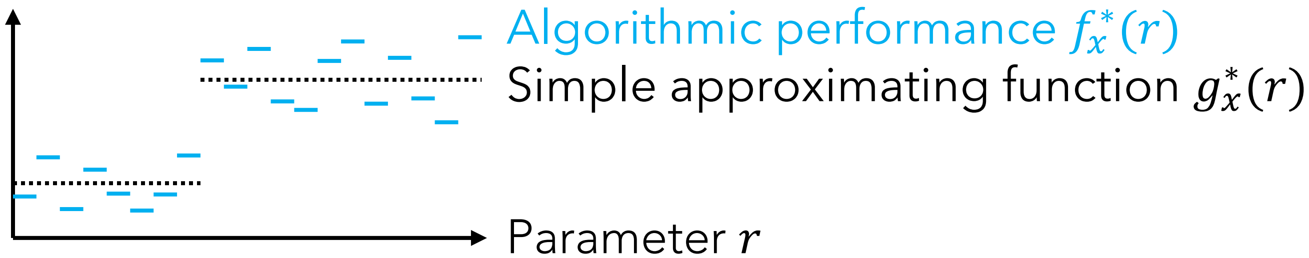
$$\left| \frac{1}{N} \sum_{i=1}^{N} f_{\boldsymbol{r}}(x_i) - \underbrace{\mathbb{E}_{x \sim \mathcal{D}}[f_{\boldsymbol{r}}(x)]}_{\textbf{Expected utility}} \right| \leq \textbf{?}$$

Main result

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

$$\left| \frac{1}{N} \sum_{x \in \mathcal{S}} f_{\boldsymbol{r}}(x) - \mathop{\mathbb{E}}_{x \sim \mathcal{D}}[f_{\boldsymbol{r}}(x)] \right|$$

$$= \tilde{O}\left(\frac{1}{N} \sum_{x \in \mathcal{S}} \|f_x^* - g_x^*\|_\infty + \text{Complexity}(\mathcal{G}) + \sqrt{\frac{1}{N}} \right)$$



Main result

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

$$\left| \frac{1}{N} \sum_{x \in \mathcal{S}} f_{\boldsymbol{r}}(x) - \mathop{\mathbb{E}}_{x \sim \mathcal{D}} [f_{\boldsymbol{r}}(x)] \right|$$

$$= \tilde{O}\left(\frac{1}{N} \sum_{x \in \mathcal{S}} \|f_x^* - g_x^*\|_\infty + \underbrace{\text{Complexity}(\mathcal{G})} + \sqrt{\frac{1}{N}} \right)$$

Measures just how "simple" the approximating functions are
Goes to 0 as N grows

Experiments

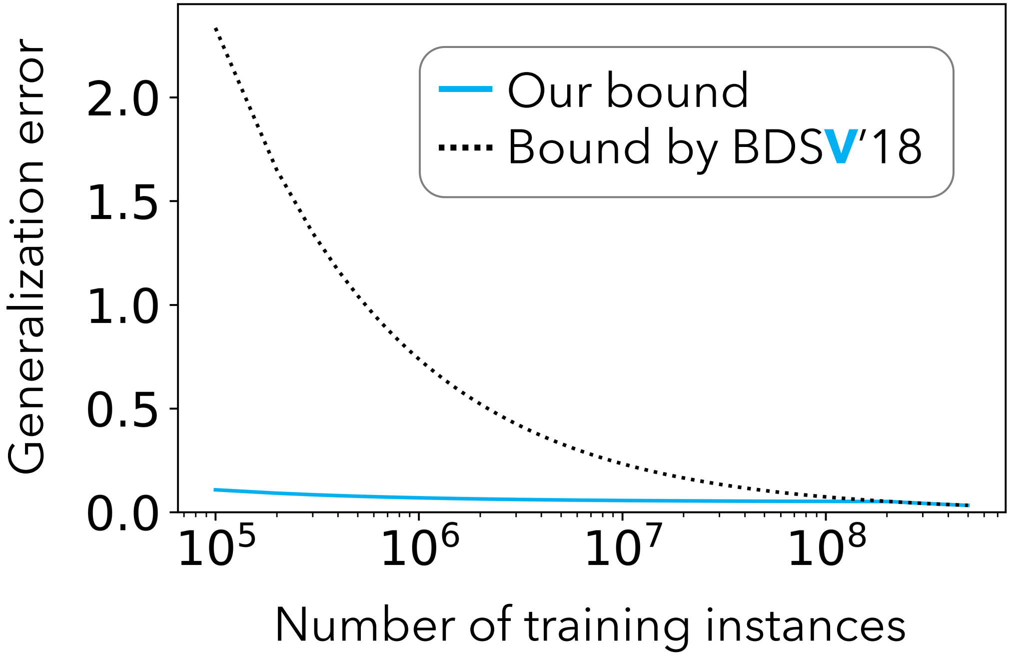
Experiments: Integer programming

Tune integer programming solver parameters

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

Distributions over auction IPs

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



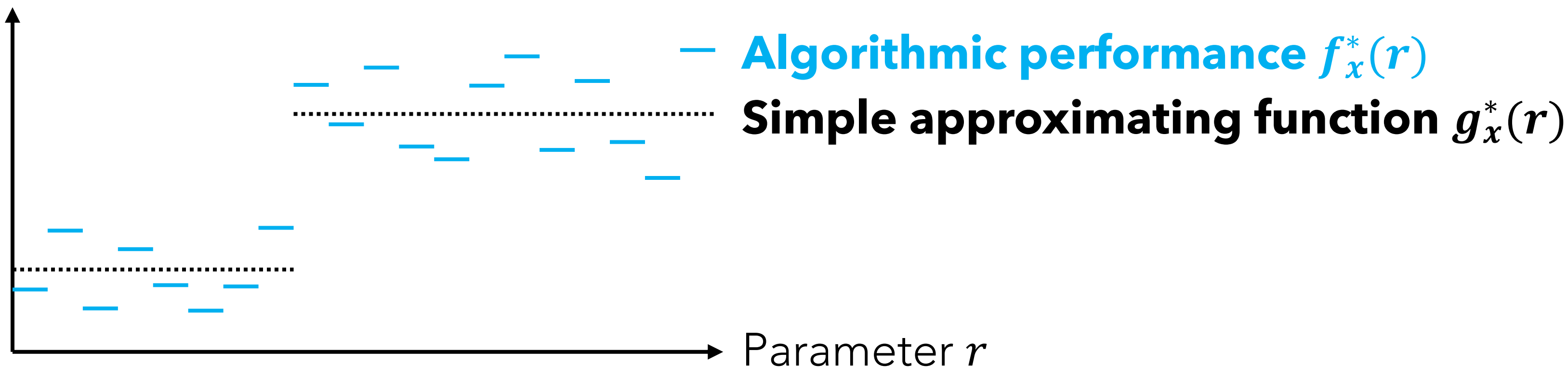
The knife-edge of dual class approximability

The knife-edge

If approximation holds under the L^∞-norm:

We provide strong guarantees

$\sup_r |f_x^*(r) - g_x^*(r)|$
is small



The knife-edge

If approximation holds under the L^∞-norm:

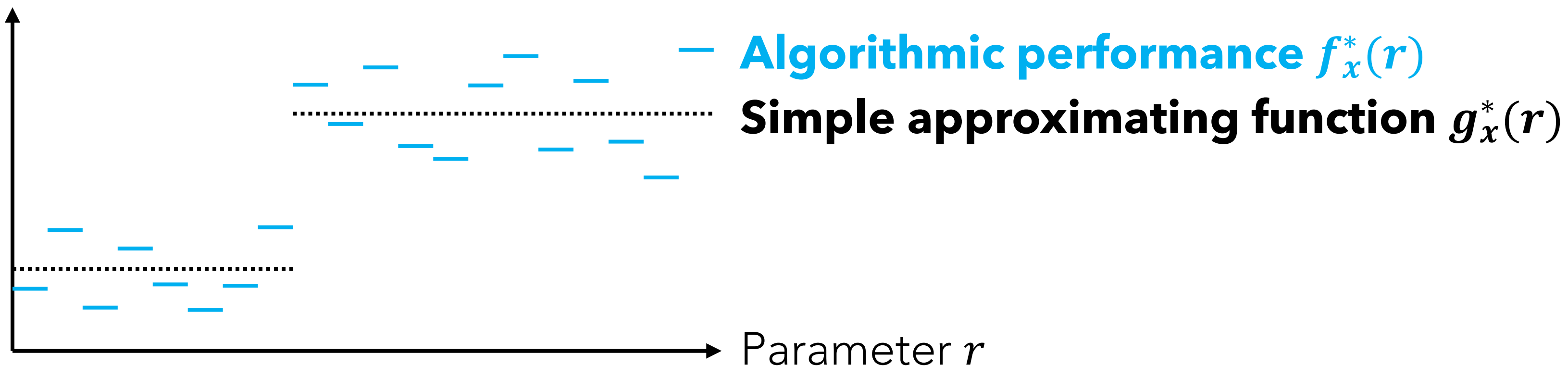
We provide strong guarantees

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

is small

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

Not possible to provide strong guarantees in worst case



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

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

Nina Balcan, Tuomas Sandholm, Ellen Vitercik

Carnegie Mellon University

INFORMS Annual Meeting 2020