Decision-Theoretic Approaches in Learning-Augmented Algorithms

Seminar GALaC

Georgii Melidi

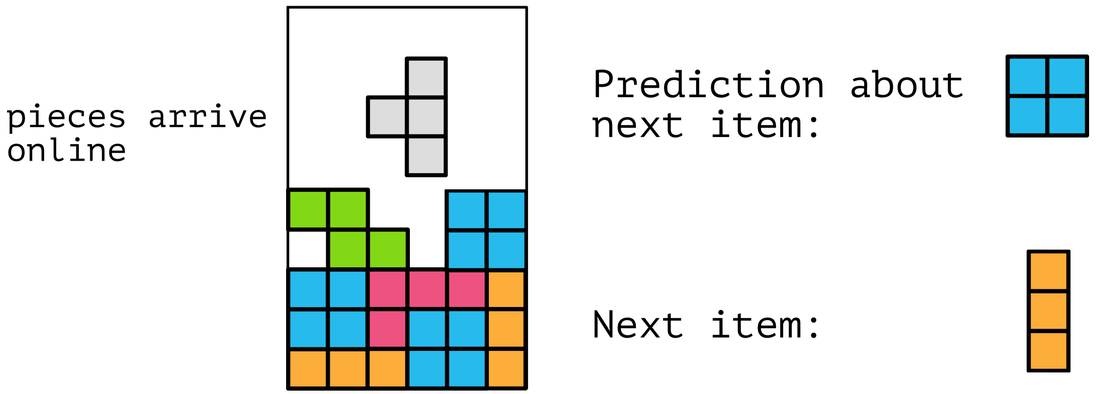
11.07.2025

Learning-augmented algorithms

Optimization under uncertainty

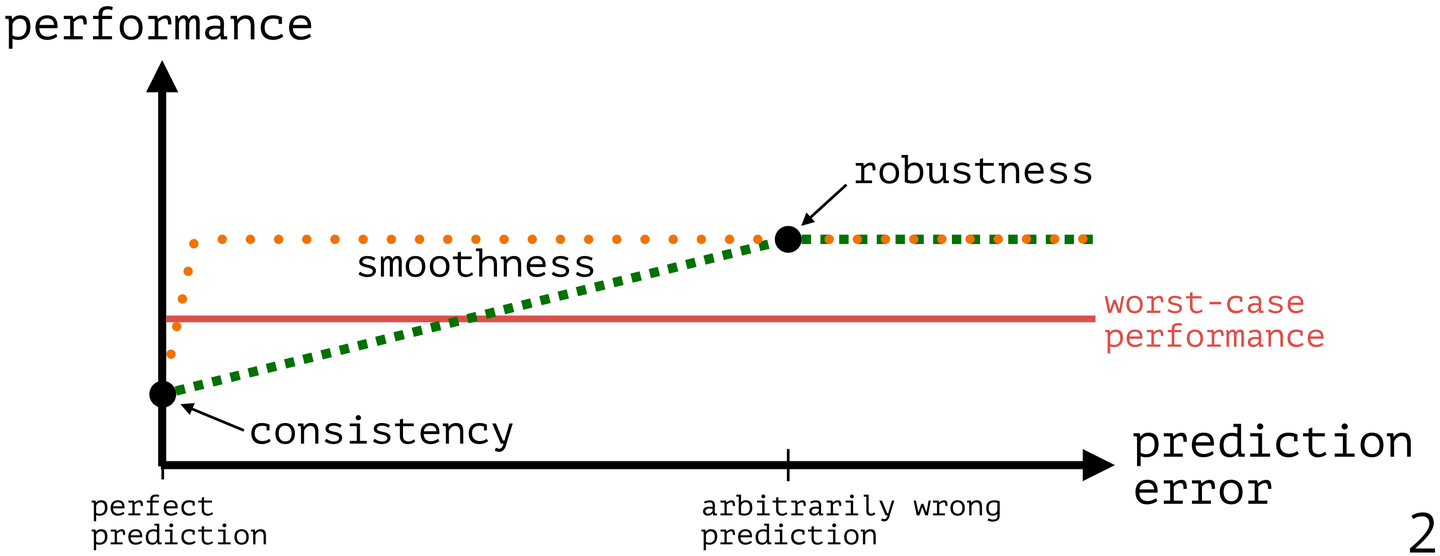
- uncertainty set
- stochastic information
- **online input**

Predictions (e.g. ML-based)



Desired algorithm properties

- Consistency
- Robustness
- Smoothness



Motivation

Pareto-optimal algorithms

✓best possible trade-off between consistency and robustness

-might be brittle

-how to select the «best» one?

Smooth algorithms

✓performance degrades gradually as the prediction error increases

✓avoid brittleness

-assumption about known upper bound on the prediction error (sometimes)

-might be suboptimal when prediction is accurate

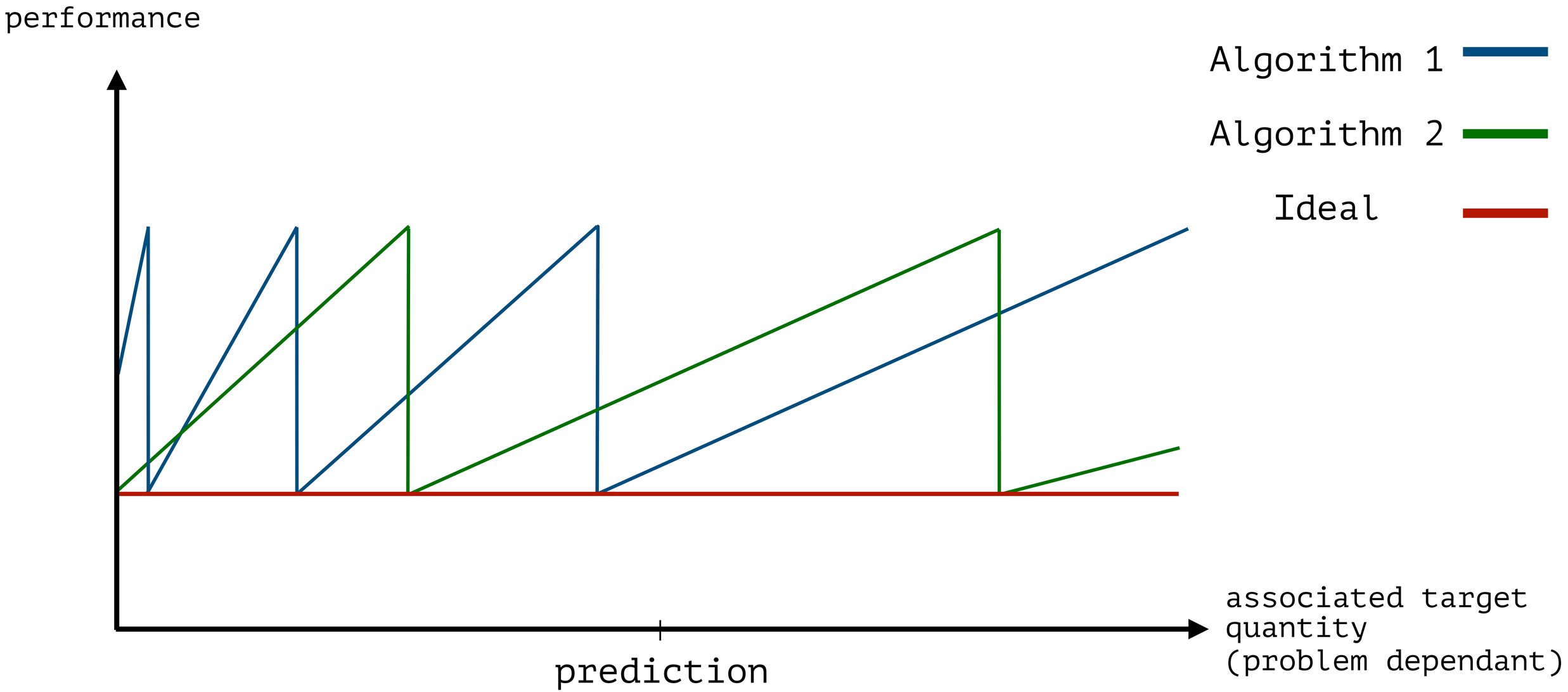
A more global approach

★aim to evaluate and compare algorithms based on how they perform across the entire range of possible errors

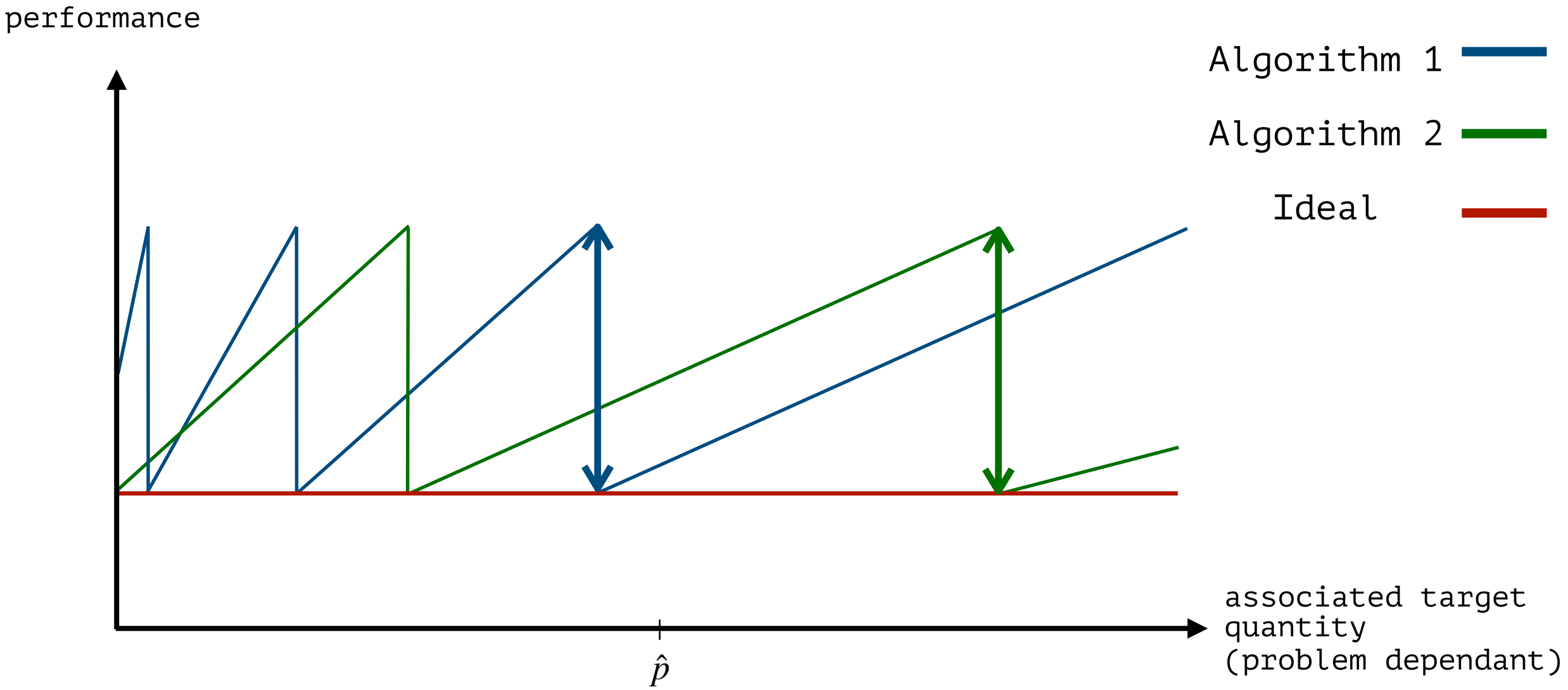
Measures:

- deterministic (distance-based)
- stochastic (risk-based)

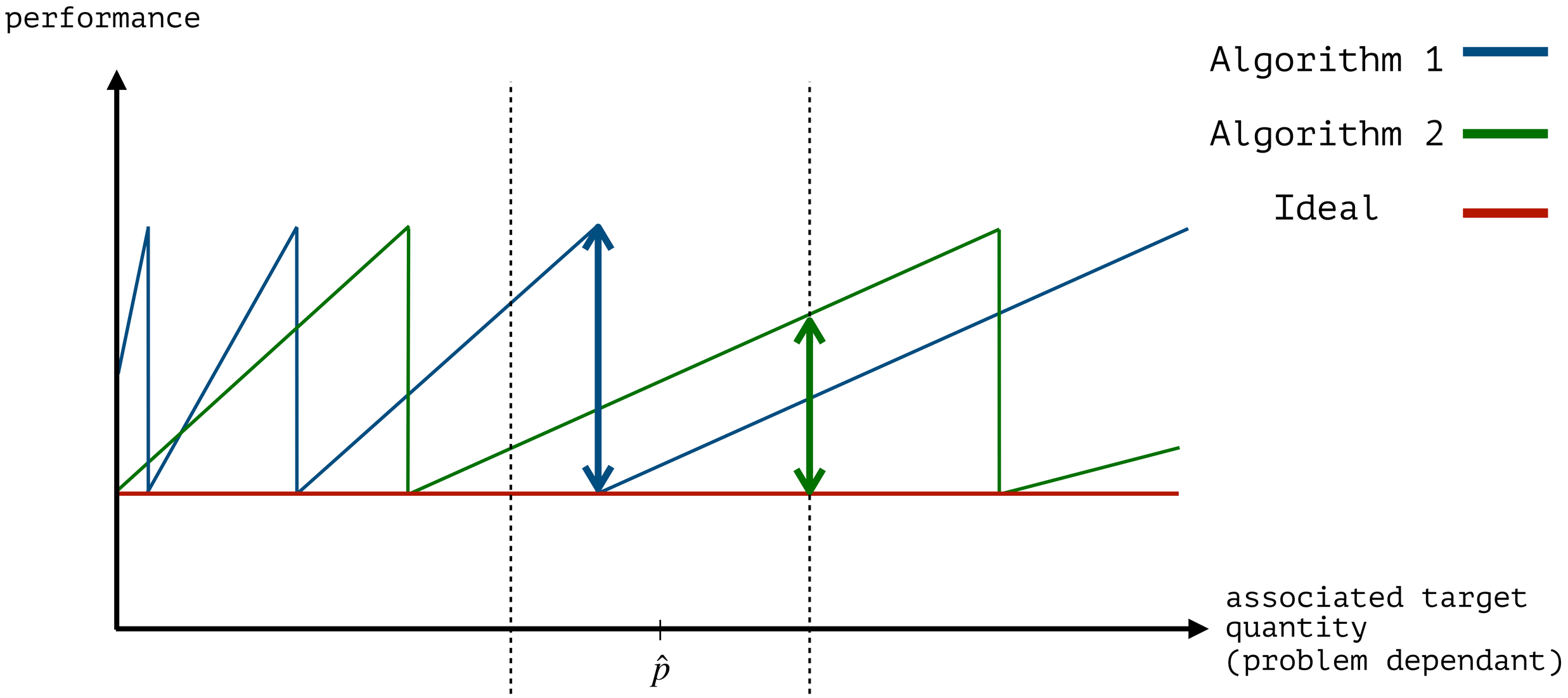
Distance-based analysis



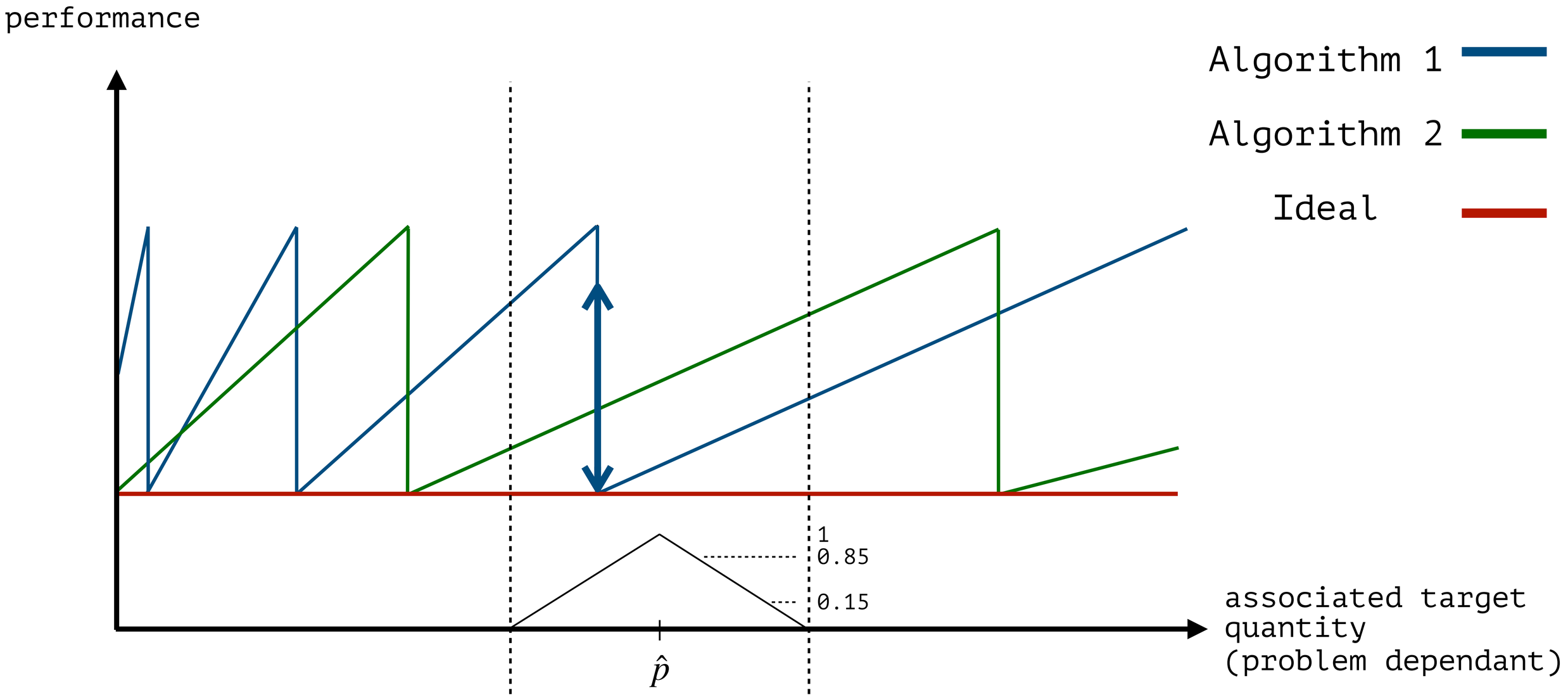
Distance-based analysis



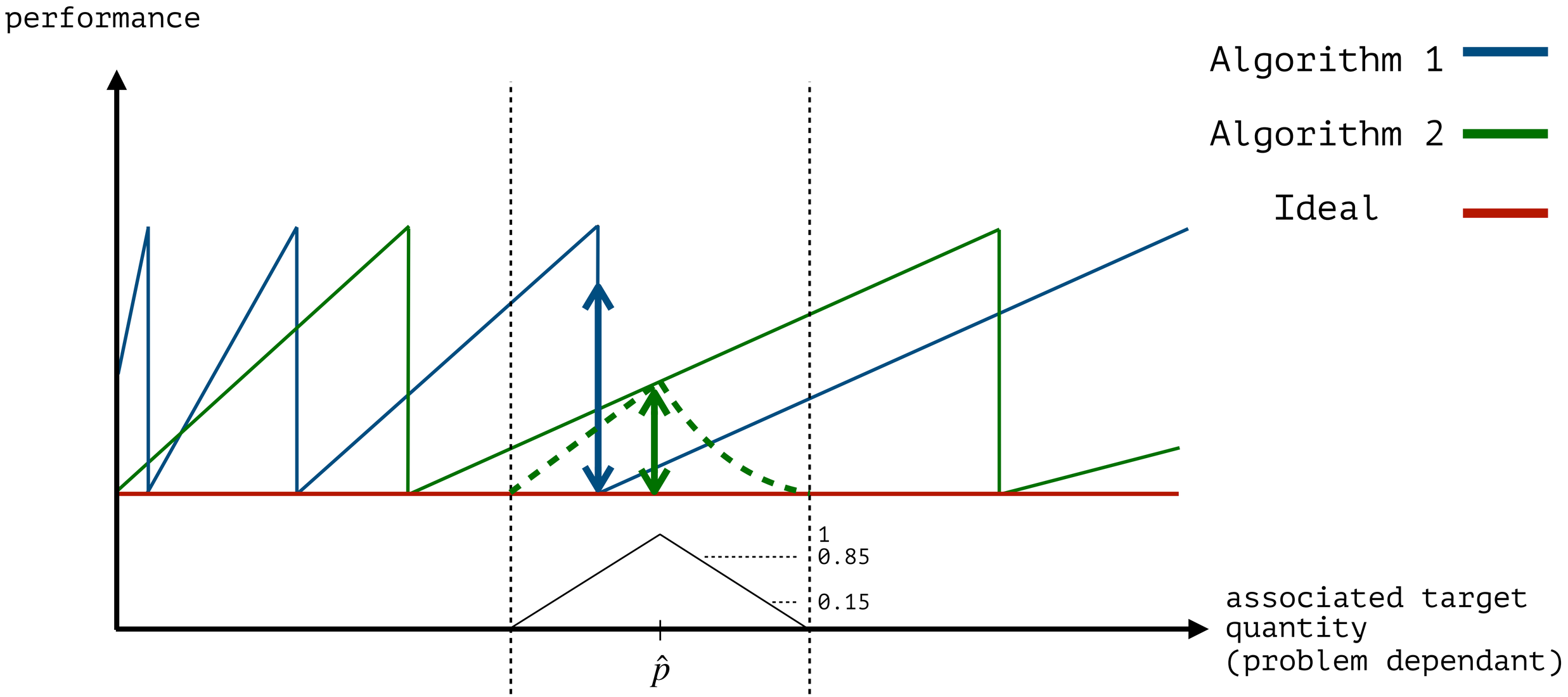
Distance-based analysis



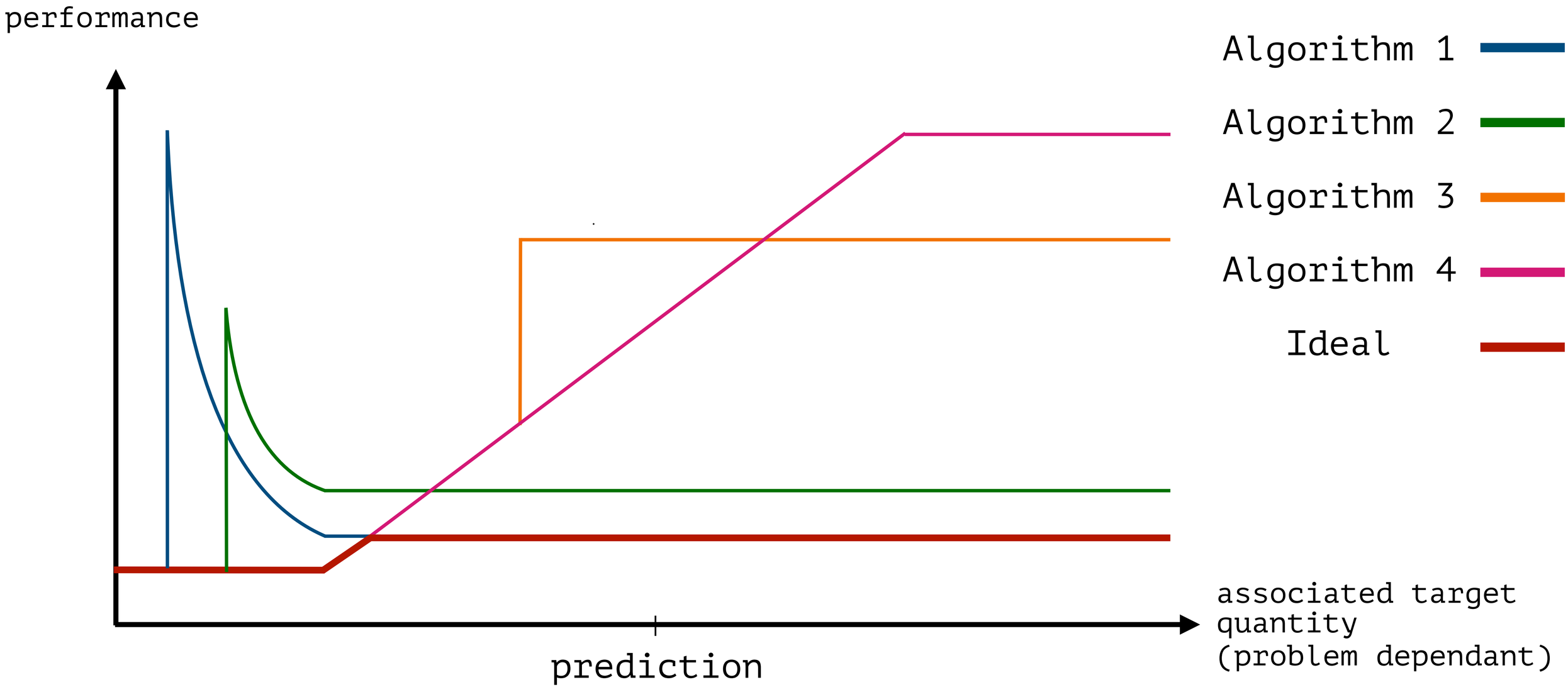
Distance-based analysis



Distance-based analysis



Distance-based analysis



Distance-based analysis

σ – input

r – robustness requirement

p^* – associated target quantity

$\hat{p}$ – prediction

$R_{\hat{p}} = [\hat{p} - h, \hat{p} + h]$ – prediction range

$w_{\hat{p}} : R_{\hat{p}} \to [0,1]$ – weight function

$A(\sigma, \hat{p})$ – cost of the algorithm A

$I_r(\sigma)$ – cost of the ideal solution

$$\mathtt{perf}(A, \sigma, \hat{p}) = \frac{A(\sigma, \hat{p})}{\mathtt{OPT}(\sigma)}; \quad \mathtt{perf}(I_r, \sigma) = \frac{I_r(\sigma)}{\mathtt{OPT}(\sigma)}$$

$$\mathtt{perf}(A, \sigma, \hat{p}) \geq \mathtt{perf}(I_r, \sigma)$$

$$d_{\mathtt{max}}(A) = \sup_{\sigma} \sup_{p \in R_{\hat{p}}} \left\{ \left(\mathtt{perf}(A, \sigma, p) - \mathtt{perf}(I_r, \sigma) \right) \cdot w_{\hat{p}}(p) \right\}$$

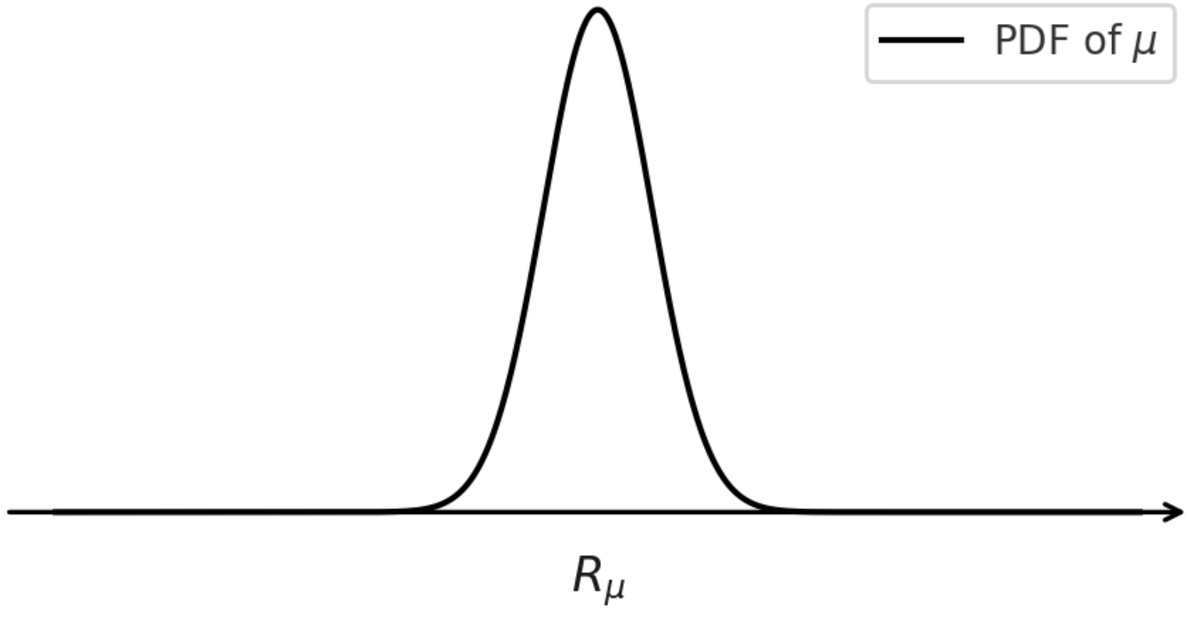
$$d_{\mathtt{avg}}(A) = \frac{1}{|R_{\hat{p}}|} \sup_{\sigma} \int_{p \in R_{\hat{p}}} \left(\mathtt{perf}(A, \sigma, p) - \mathtt{perf}(I_r, \sigma) \right) \cdot w_{\hat{p}}(p)\, dp$$

Risk-based analysis

$$\mathrm{CVaR}_\alpha(A(\sigma)) = \inf_t \left\{ t + \frac{1}{1-\alpha}\mathbb{E}[(A(\sigma)-t)^+] \right\},$$

where $\alpha \in [0,1)$.

prediction is in the form of a distribution μ, with support over an interval R_μ



$\alpha = 0$ - risk-seeking algorithm that aims to maximize its expected profit

$\alpha \to 1$ - risk averse algorithm

[] **Nicolas Christianson, Bo Sun, Steven H. Low, and Adam Wierman.** Risk-Sensitive Online Algorithms. **2024.**

Risk-based analysis

For an algorithm A, and given $\alpha \in [0,1)$, we define the α-consistency:

$$\alpha - \text{cons}(A) = \sup_{F \in \mathcal{F}} \left(\frac{\text{CVaR}_{\alpha,F}(A(\sigma))}{\mathbb{E}_{\sigma \sim F}[\text{OPT}(\sigma)]} \right)$$

This measure is a risk-inclusive generalization of consistency, and interpolates between two extreme cases:

$\alpha = 0$ - risk-seeking algorithm

$$\text{CVaR}_{\alpha,F}(A) = \mathbb{E}_{\sigma \sim F}[A(\sigma)] \text{ [*]}$$

$\alpha \to 1$ - risk averse algorithm

$$\text{CVaR}_{\alpha,F}(A) = \inf_{\sigma \in \text{supp}(F)} A(\sigma)$$

[*] **Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, Ali Vakilian, and Nikos Zarifis.** Learning Online Algorithms with DistributionalAdvice. **2021**

Ski Rental

Ski Rental

x – unknown number of skiing days

b – buying cost

For an algorithm A_T that buys at time T, the total cost is x if $x < T$, and $b + T$ if $x \geq T$.

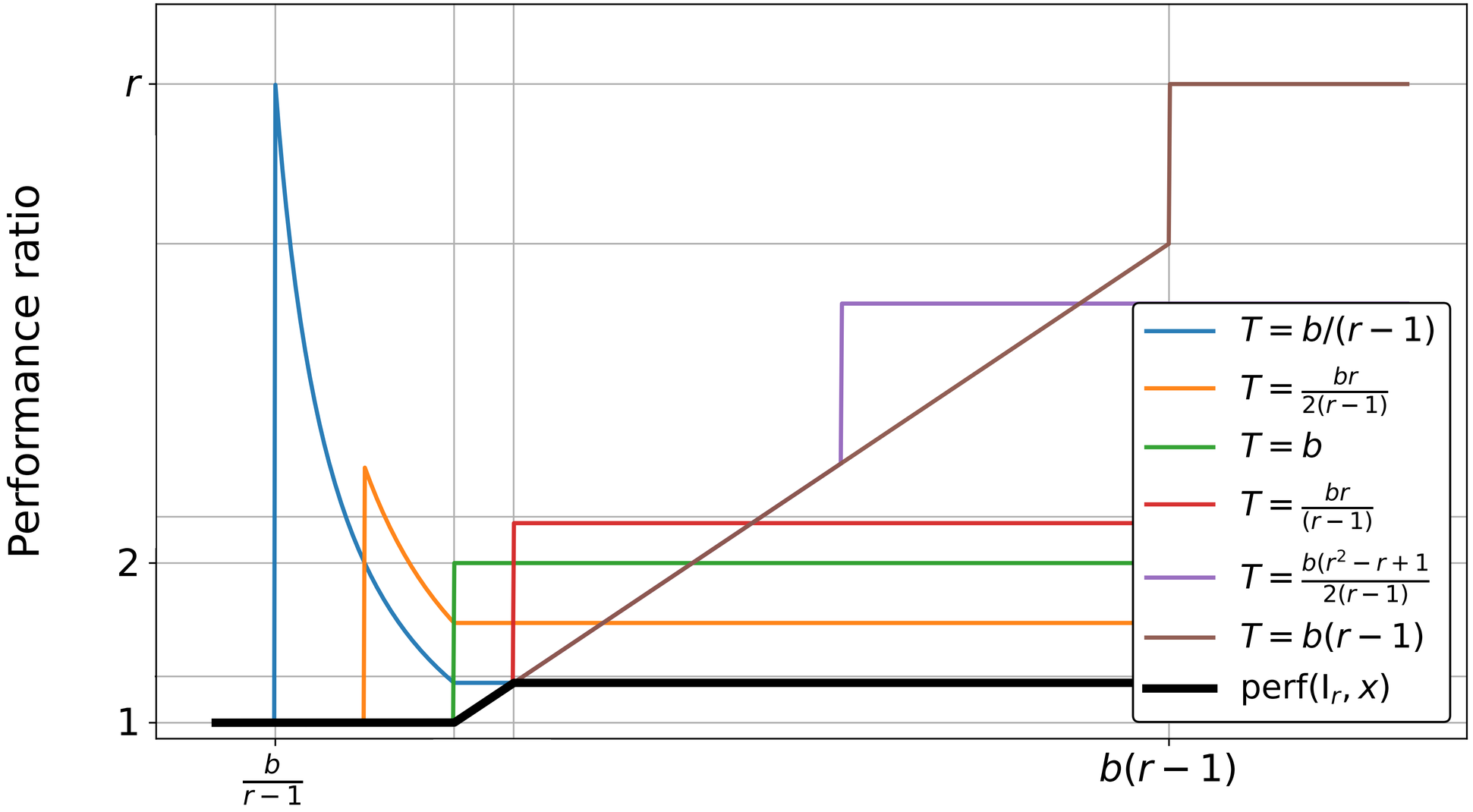
The best deterministic competitive ratio is equal to 2.

For any $r \geq 2$, an r-robust algorithm must purchase within $[b/(r-1), b(r-1)]$.

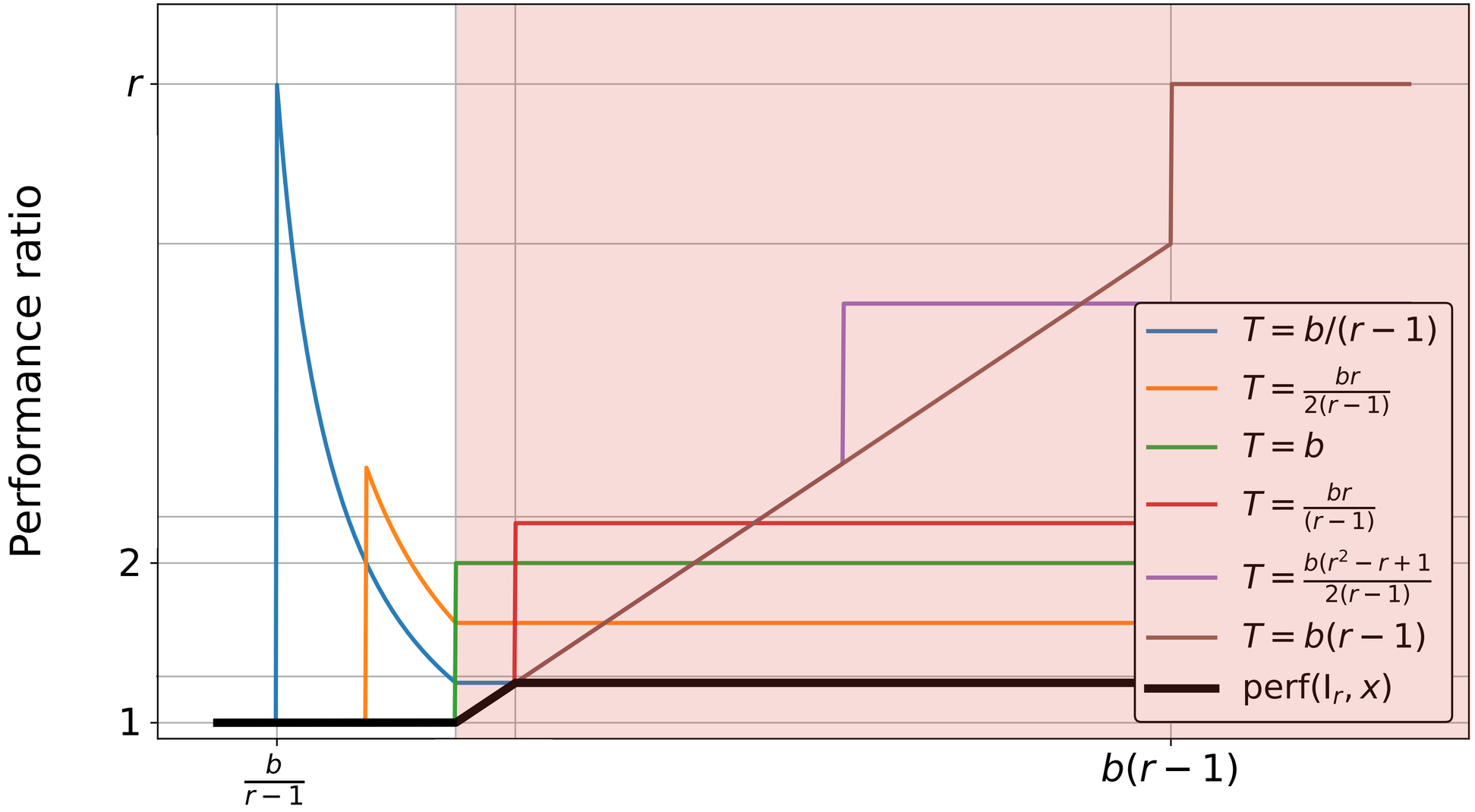
Objective

Given a robustness requirement r and a prediction y on the number of skiing days, find the optimal $T \in [b/(r-1), b(r-1)]$ such that A_T minimizes the distance-based objective (d_{max} or d_{avg}) or risk-based objective (CVAR).

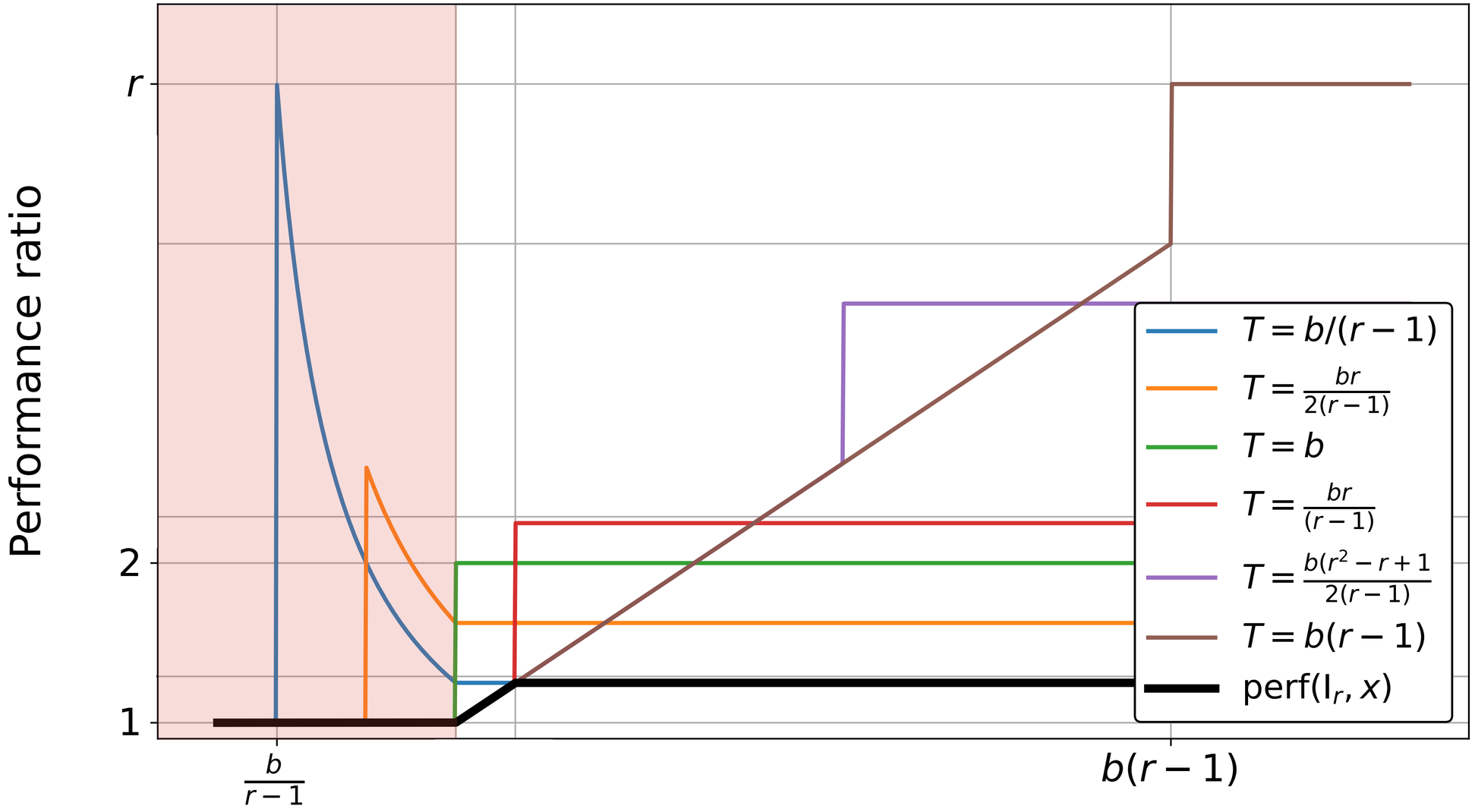
Ski Rental


Performance ratio
r
2
1
$\frac{b}{r-1}$
$b(r-1)$
Ski season length
$T = b/(r-1)$
$T = \frac{br}{2(r-1)}$
$T = b$
$T = \frac{br}{(r-1)}$
$T = \frac{b(r^2-r+1}{2(r-1)}$
$T = b(r-1)$
perf(I_r, x)

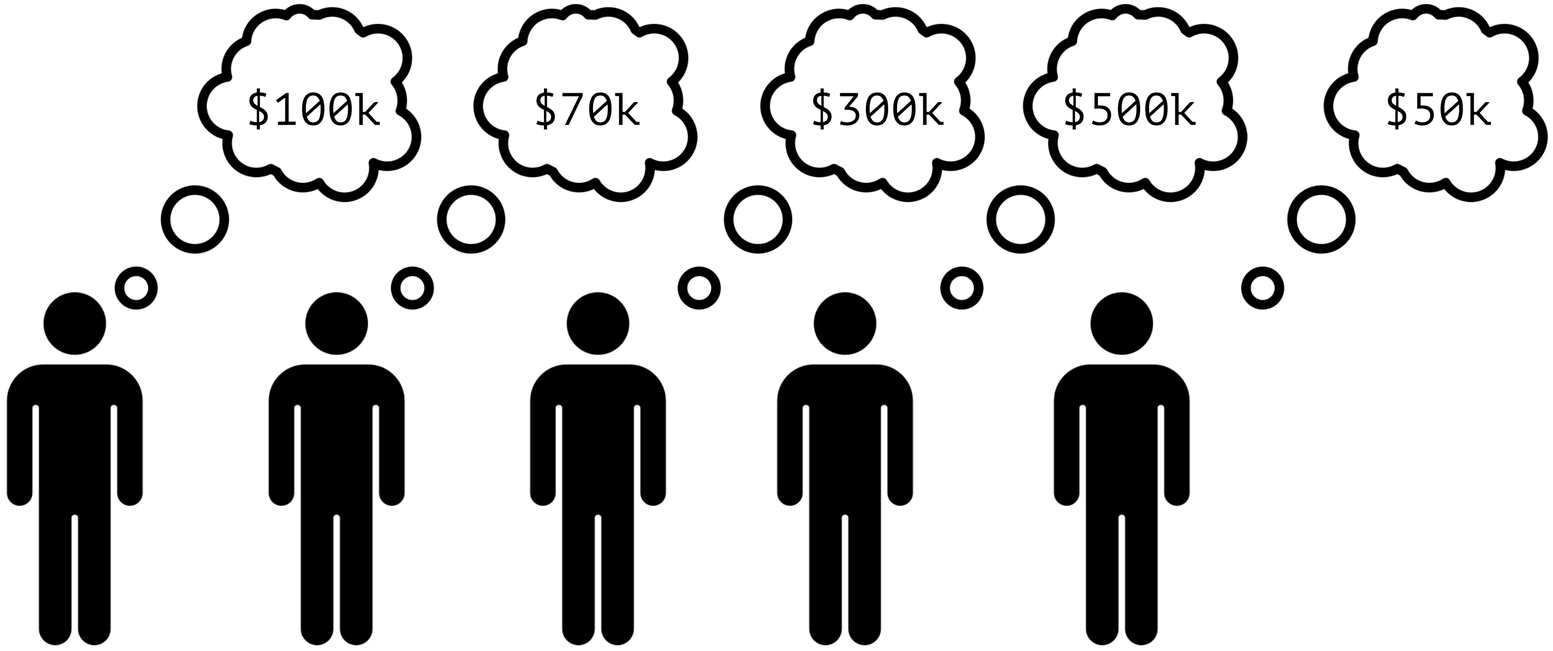
Ski Rental


Performance ratio
r
2
1
b/(r−1)
b(r−1)
Ski season length
T = b/(r − 1)
T = br/2(r − 1)
T = b
T = br/(r − 1)
T = b(r² − r + 1)/2(r − 1)
T = b(r − 1)
perf(I_r, x)

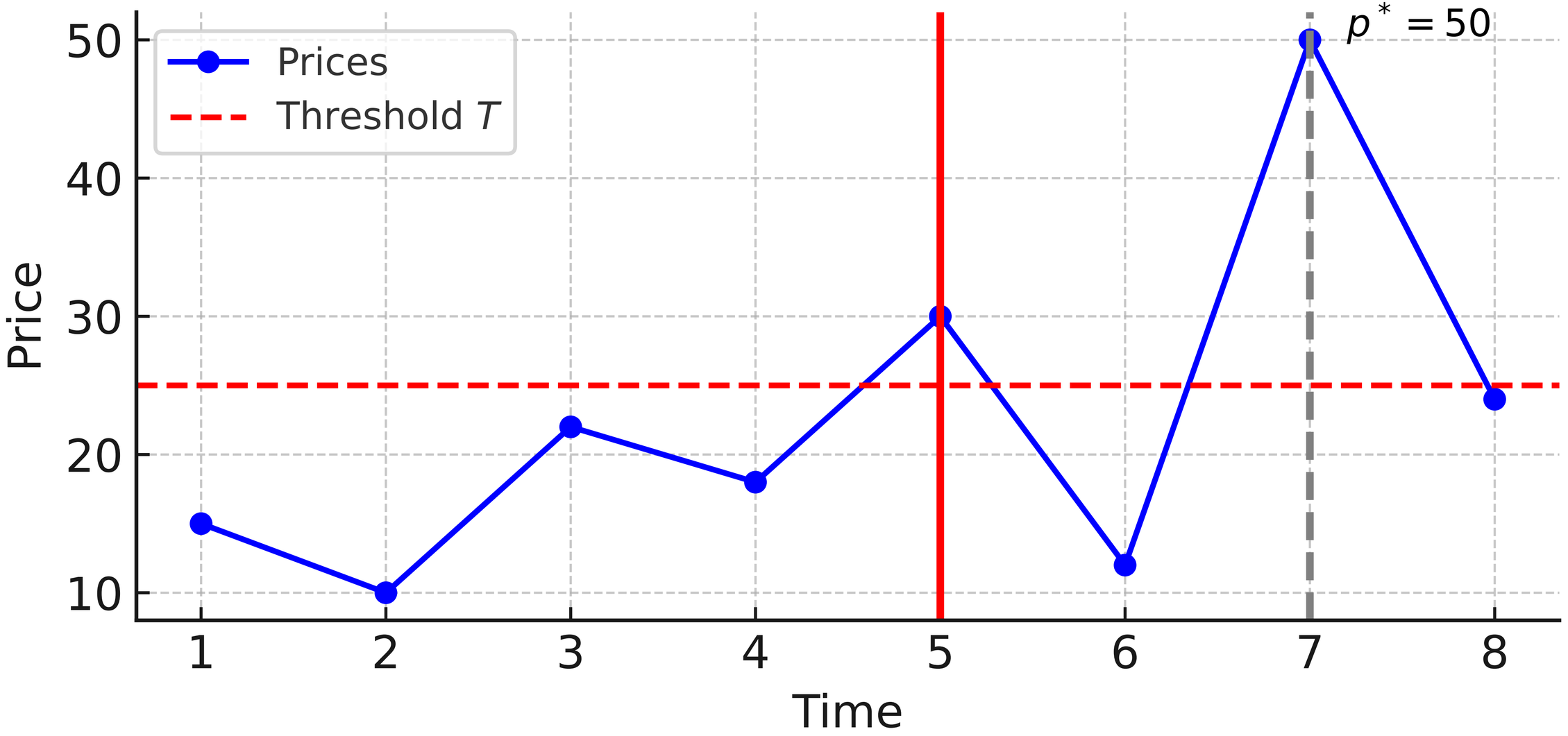
Ski Rental


Performance ratio
r
2
1
$\frac{b}{r-1}$
$b(r-1)$
Ski season length
$T = b/(r-1)$
$T = \frac{br}{2(r-1)}$
$T = b$
$T = \frac{br}{(r-1)}$
$T = \frac{b(r^2-r+1}{2(r-1)}$
$T = b(r-1)$
perf(I_r, x)

1-Max Search



1-Max Search



1-Max Search

Input σ - sequence of prices in $[1,M]$

A_T - threshold algorithm with $T \in [1,M]$

p^* - max price

$\hat{p}$ - prediction

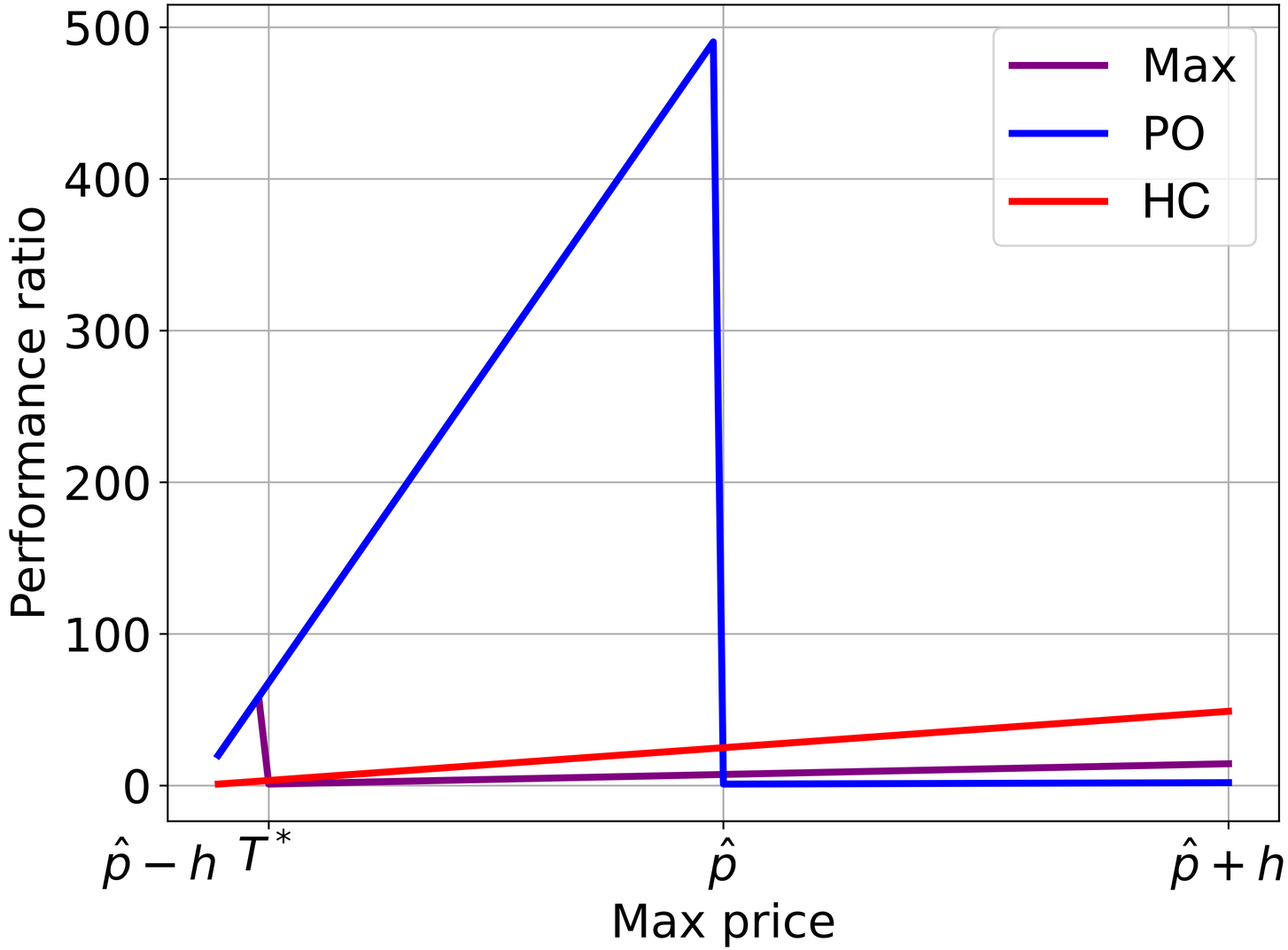
For any $r \geq \sqrt{M}$, algorithm A_T is r-robust if and only if $T \in [M/r, r]$.

Optimal competitive ratio without prediction is $\sqrt{M}$.

Objective

For each of the decision-theoretic models and a given robustness requirement r, find the optimal threshold $T \in [M/r, r]$ that optimizes the corresponding measure.

1-Max Search



	MAX	PO	HC
Avg Ratio	9.26	127.08	24.9

Time vs PO	45.69
Time vs HC	96.04

1-Max Search|Real-World Experiments

Setting

- EUR exchange rates against CHF, USD, GBP, and JPY (6672 prices over 25 years).
- Bitcoin (USD) price data recorded every minute from 2020 to 2024, totalling 2,630,880 prices.

For each currency, we run 10,000 trials and computed the average empirical competitive ratios.

1-Max Search

Currency	PO	HC	Max	Avg	CVaR
CHF	1.4853	1.5261	1.2611	1.2709	1.3328
GBP	1.4018	1.1480	1.7115	1.7134	1.4249
JPY	1.5115	1.0846	1.0869	1.0869	1.1650
USD	1.5229	1.3561	1.2339	1.2386	1.3012
BTC	14.9876	8.9721	8.7997	8.7643	7.0627

Conclusion

- Introduced new evaluation metrics that enable the optimization of learning-augmented algorithms over the entire range of prediction error.
- Designed theoretically optimal algorithms for several well-studied problems (**1-max search**, **ski rental**, **contract scheduling**) and demonstrated that they outperform the known, extreme case approaches in practice.

Future work

- Extension to other learning-augmented problems, such as **knapsack problem** and **secretary problem**.
- Consider problems with multi-valued predictions such as **packing problems**.

Conclusion

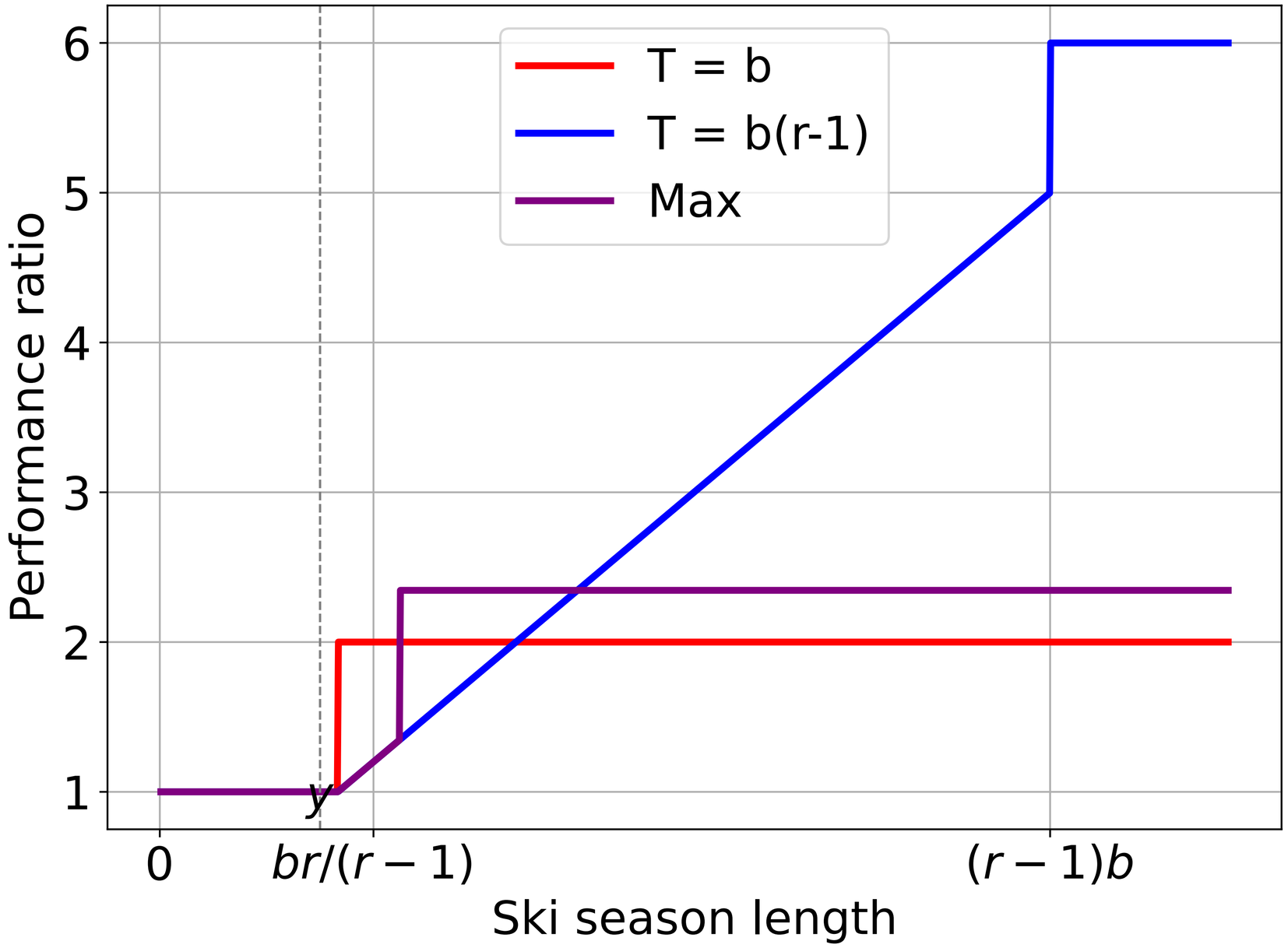
- Introduced new evaluation metrics that enable the optimization of learning-augmented algorithms over the entire range of prediction error.
- Designed theoretically optimal algorithms for several well-studied problems (**1-max search**, **ski rental**, **contract scheduling**) and demonstrated that they outperform the known, extreme case approaches in practice.

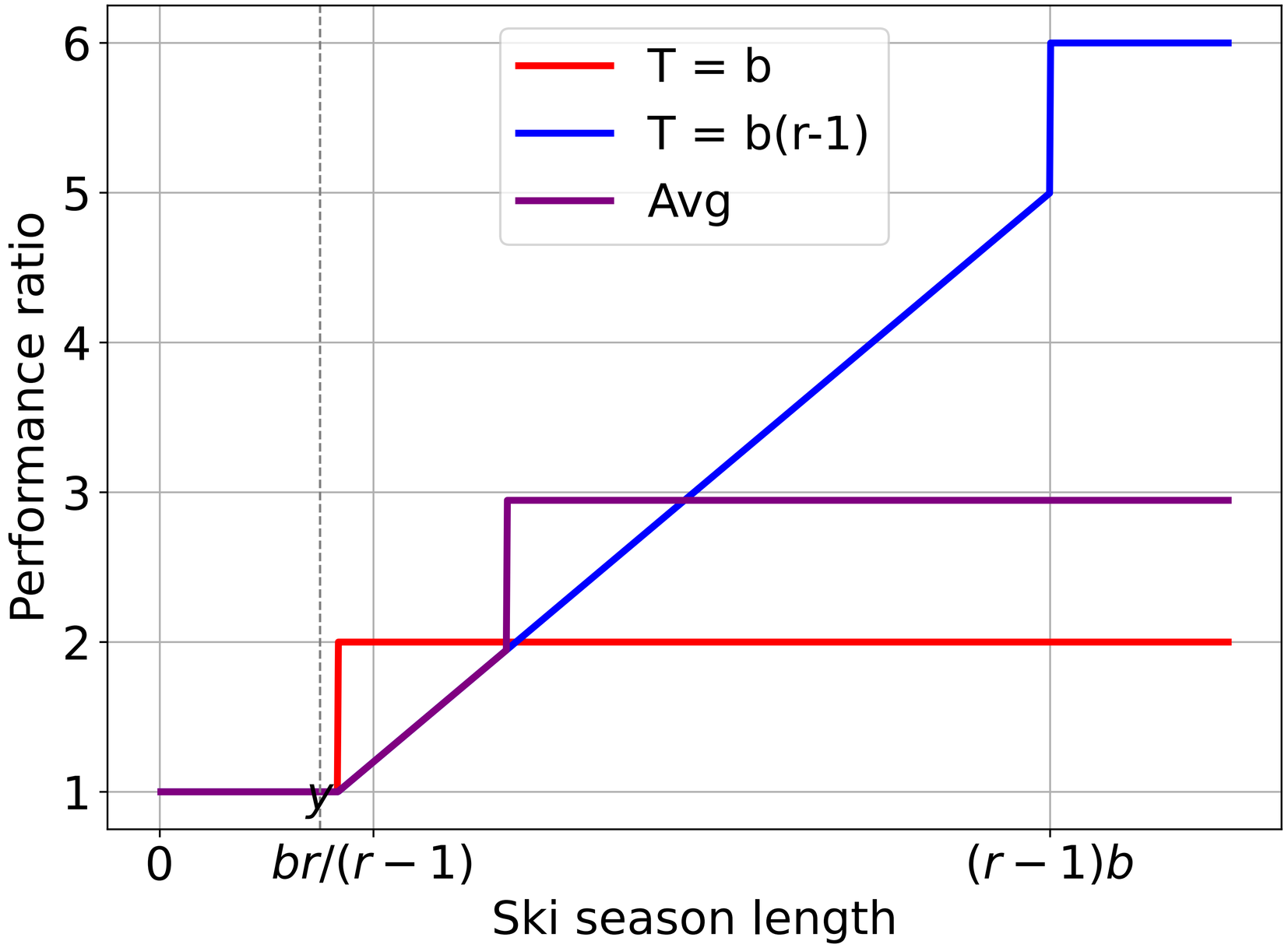
Future work

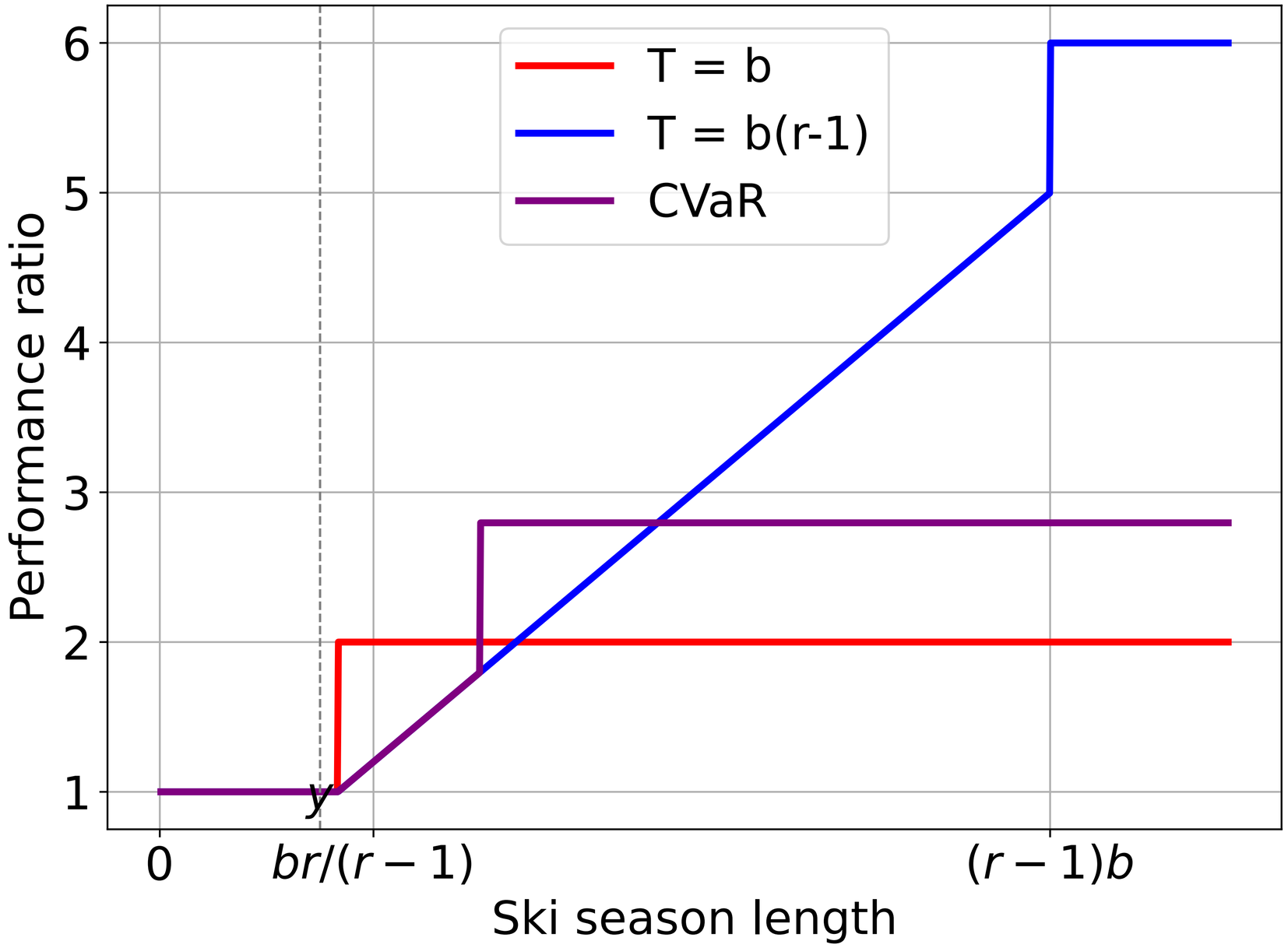
- Extension to other learning-augmented problems, such as **knapsack problem** and **secretary problem**.
- Consider problems with multi-valued predictions such as **packing problems**.

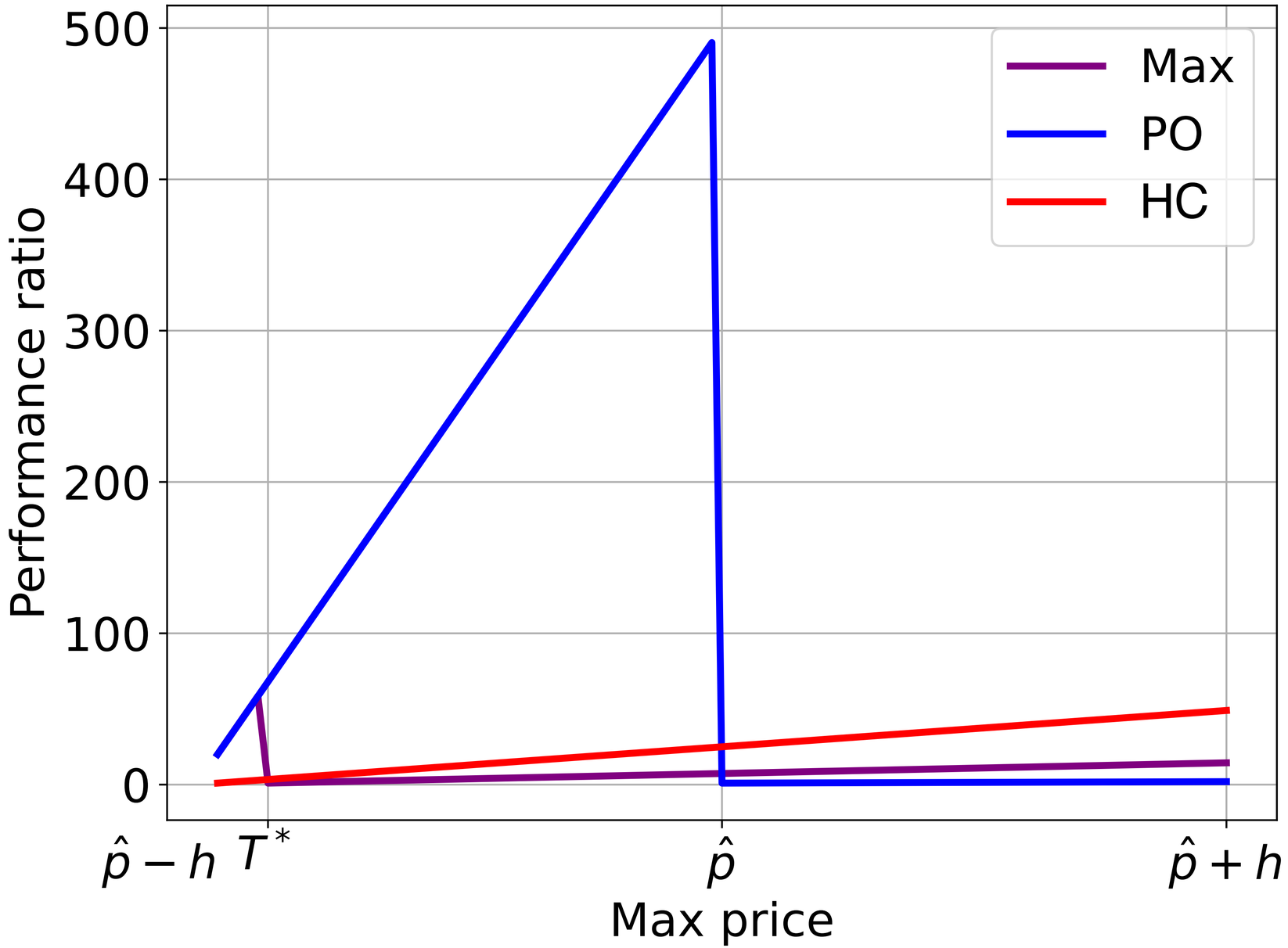
Thank you!

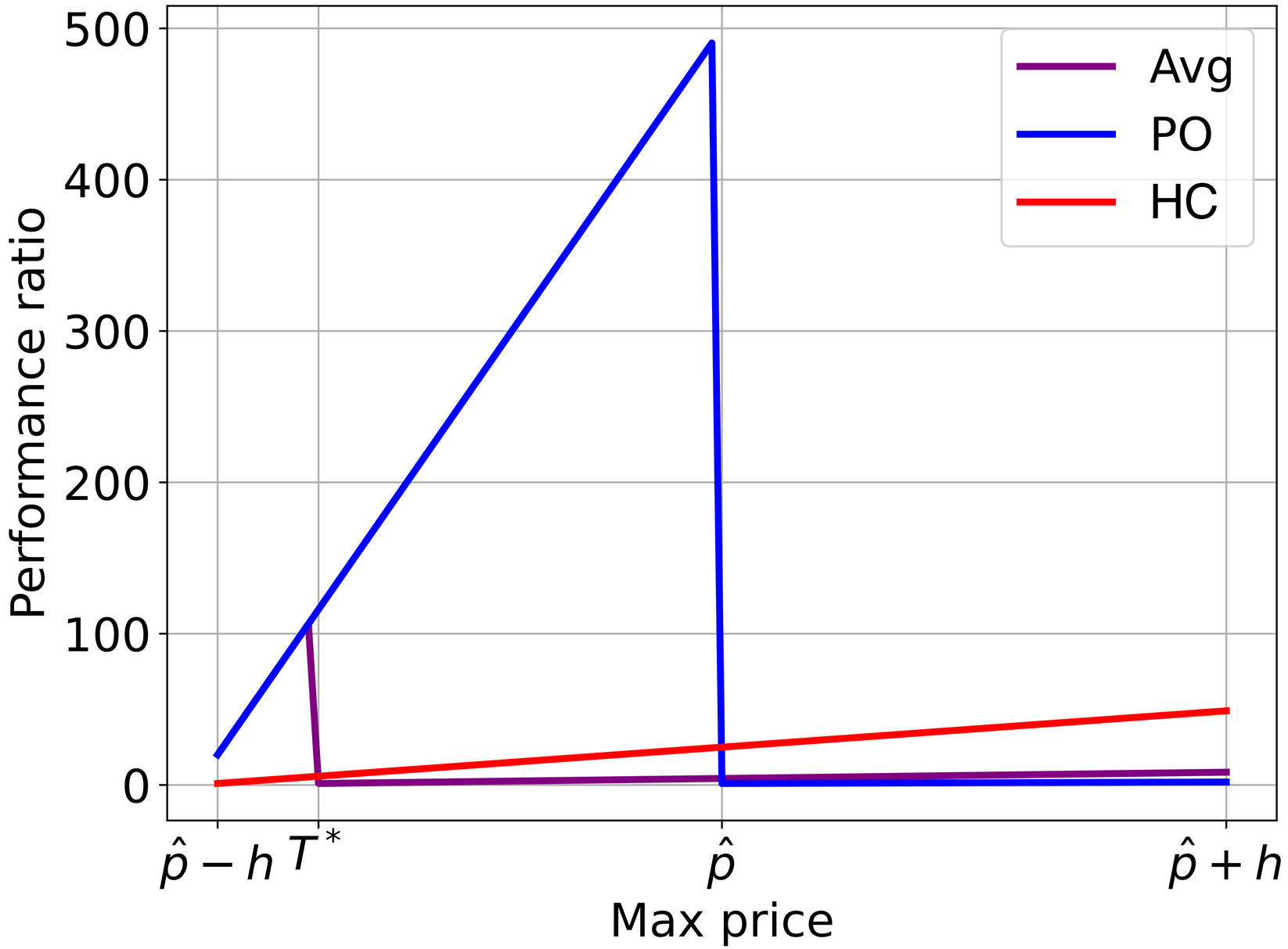
STOP

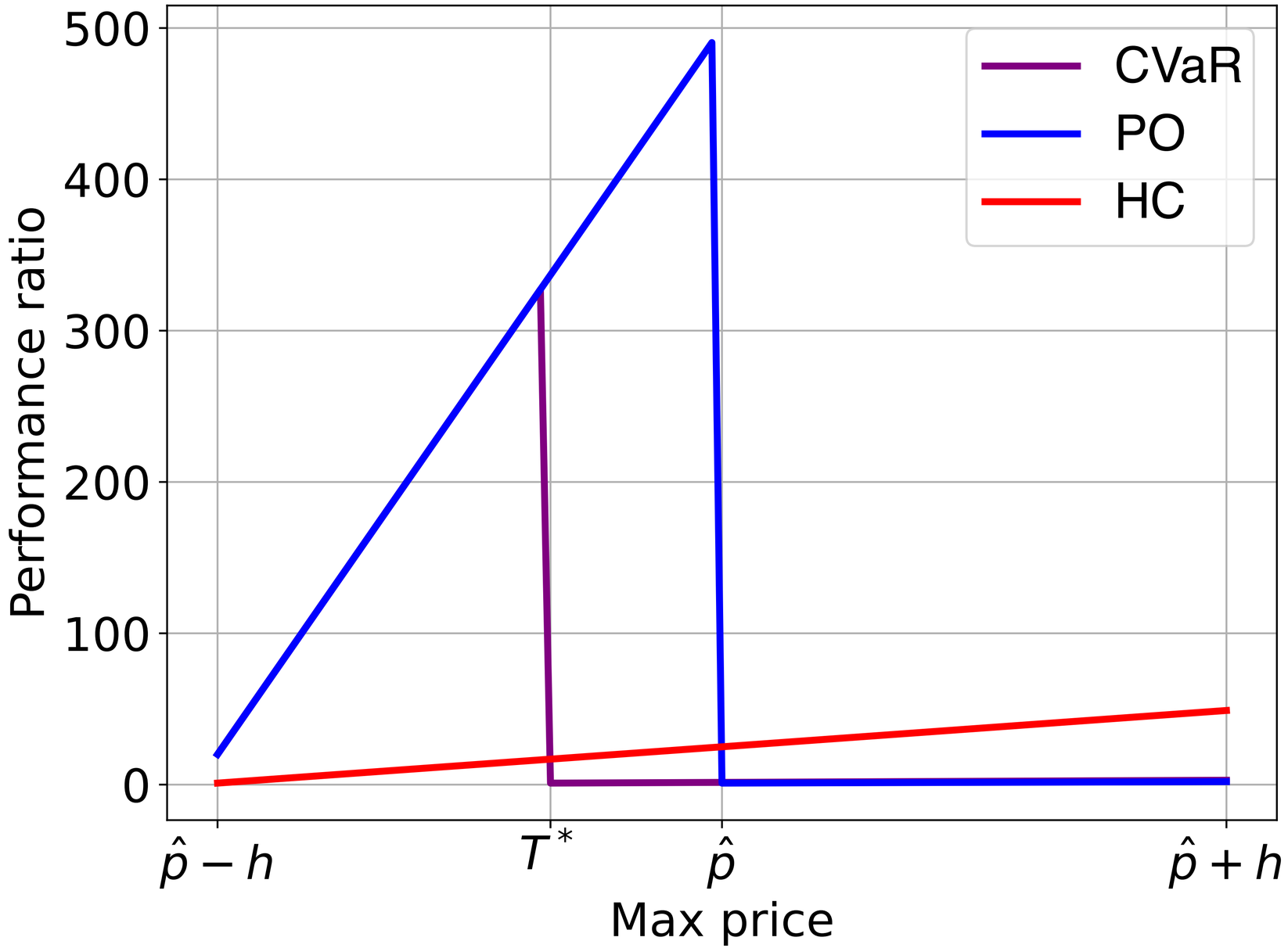

T = b
T = b(r-1)
Max
Performance ratio
Ski season length
1
2
3
4
5
6
0
$br/(r-1)$
$(r-1)b$
y


T = b
T = b(r-1)
Avg
Performance ratio
Ski season length
1
2
3
4
5
6
0
$br/(r-1)$
$(r-1)b$
y


T = b
T = b(r-1)
CVaR
Performance ratio
Ski season length
1
2
3
4
5
6
0
$br/(r-1)$
$(r-1)b$
y


500
400
300
200
100
0
Performance ratio
Max
PO
HC
$\hat{p}-h$
T^*
$\hat{p}$
$\hat{p}+h$
Max price


500
400
300
200
100
0
Performance ratio
Avg
PO
HC
$\hat{p} - h$
T^*
$\hat{p}$
$\hat{p} + h$
Max price


500
400
300
200
100
0
Performance ratio
CVaR
PO
HC
$\hat{p} - h$
T^*
$\hat{p}$
$\hat{p} + h$
Max price