Contract Scheduling with Distributional and Multiple Advice

Spyros Angelopoulos Marcin Bienkowski Christoph Dürr
Bertrand Simon

IJCAI 2024

CNRS / CC-IN2P3

COA Workshop, Nov. 2024

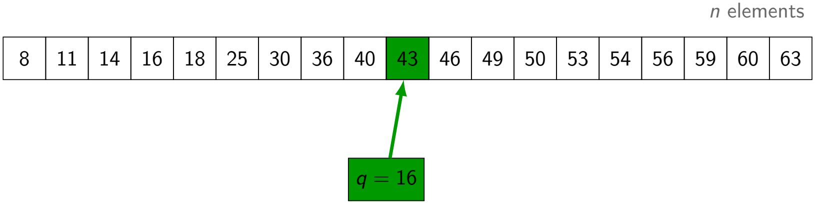
Algorithm with predictions example: binary search

n elements

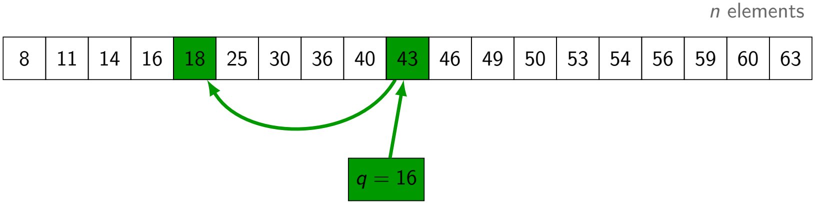
8	11	14	16	18	25	30	36	40	43	46	49	50	53	54	56	59	60	63

$q = 16$

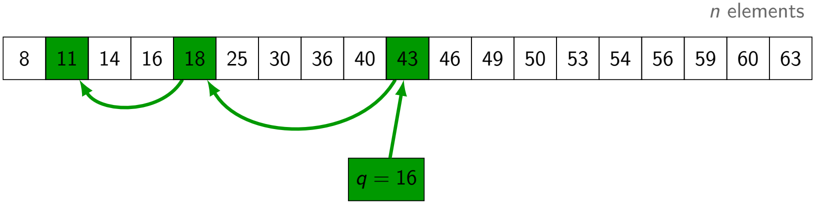
Algorithm with predictions example: binary search



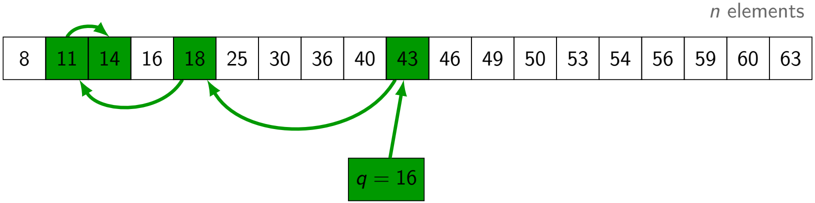
Algorithm with predictions example: binary search



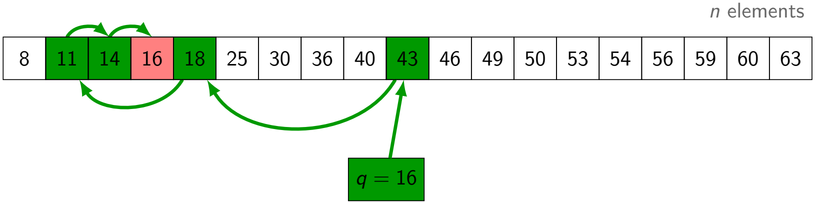
Algorithm with predictions example: binary search



Algorithm with predictions example: binary search



Algorithm with predictions example: binary search



Algorithm with predictions example: binary search

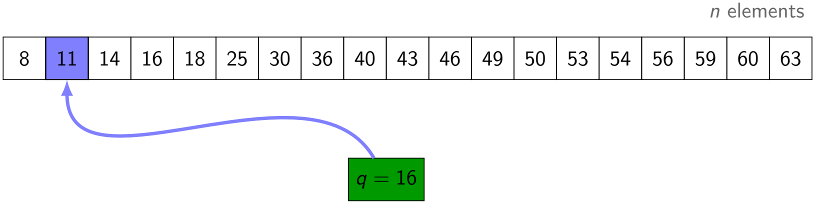
n elements

8	11	14	16	18	25	30	36	40	43	46	49	50	53	54	56	59	60	63

$q = 16$

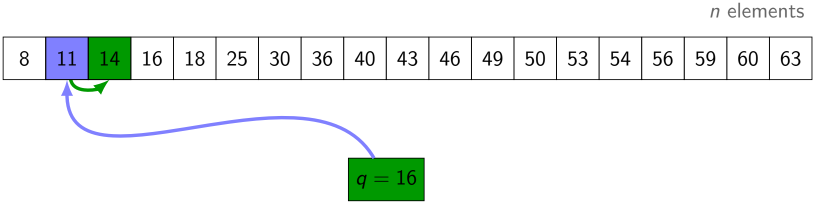
Prediction: position $h(q)$ Error: $\eta = |h(q) - index(q)|$

Algorithm with predictions example: binary search



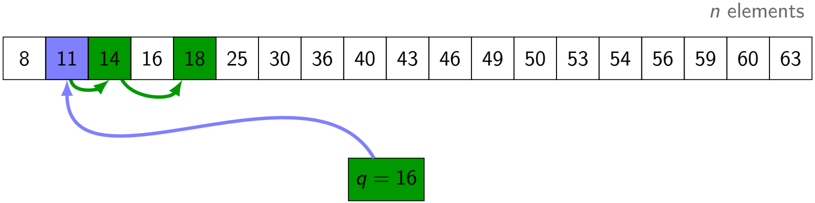
Prediction: position $h(q)$ Error: $\eta = |h(q) - index(q)|$

Algorithm with predictions example: binary search



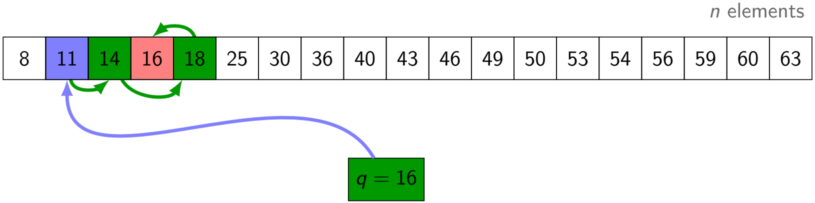
Prediction: position $h(q)$ Error: $\eta = |h(q) - index(q)|$

Algorithm with predictions example: binary search



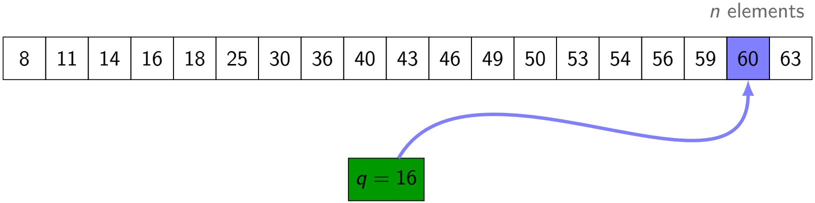
Prediction: position $h(q)$ Error: $\eta = |h(q) - index(q)|$

Algorithm with predictions example: binary search



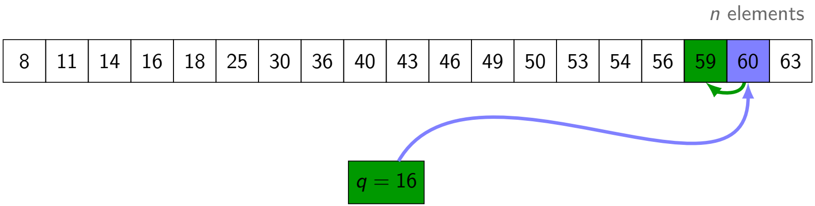
Prediction: position $h(q)$ Error: $\eta = |h(q) - index(q)|$

Algorithm with predictions example: binary search



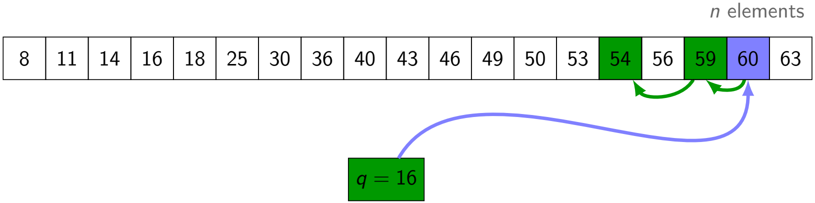
Prediction: position $h(q)$ Error: $\eta = |h(q) - index(q)|$

Algorithm with predictions example: binary search



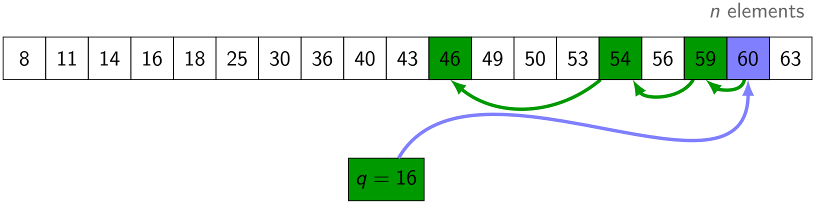
Prediction: position $h(q)$ Error: $\eta = |h(q) - index(q)|$

Algorithm with predictions example: binary search



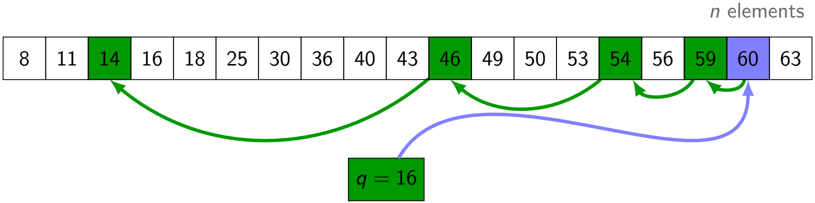
Prediction: position $h(q)$ Error: $\eta = |h(q) - index(q)|$

Algorithm with predictions example: binary search



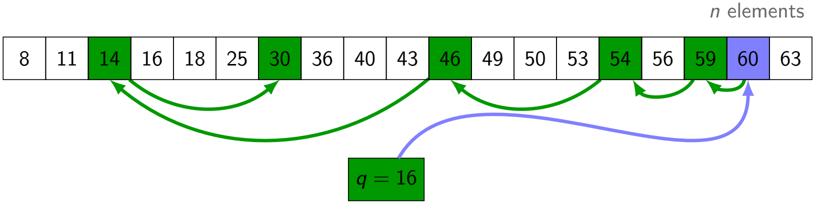
Prediction: position $h(q)$ Error: $\eta = |h(q) - index(q)|$

Algorithm with predictions example: binary search



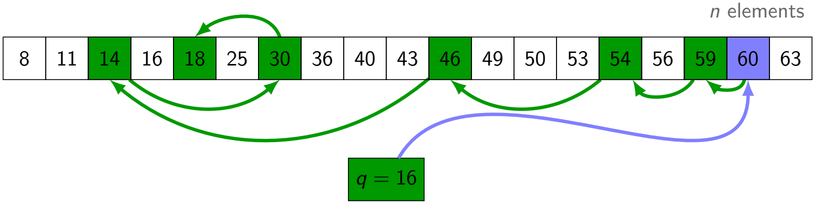
Prediction: position $h(q)$ Error: $\eta = |h(q) - index(q)|$

Algorithm with predictions example: binary search



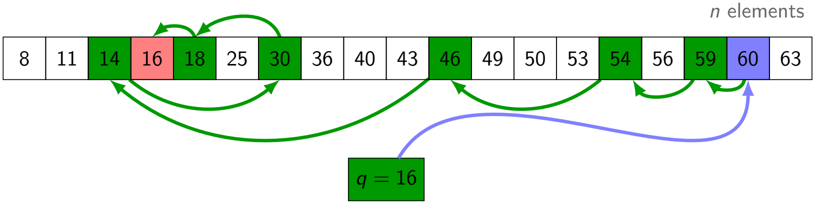
Prediction: position $h(q)$ Error: $\eta = |h(q) - index(q)|$

Algorithm with predictions example: binary search



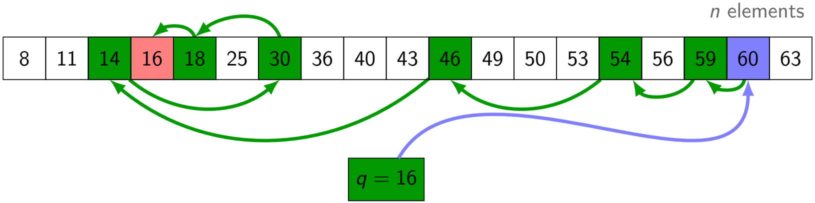
Prediction: position $h(q)$ Error: $\eta = |h(q) - index(q)|$

Algorithm with predictions example: binary search



Prediction: position $h(q)$ Error: $\eta = |h(q) - index(q)|$

Algorithm with predictions example: binary search

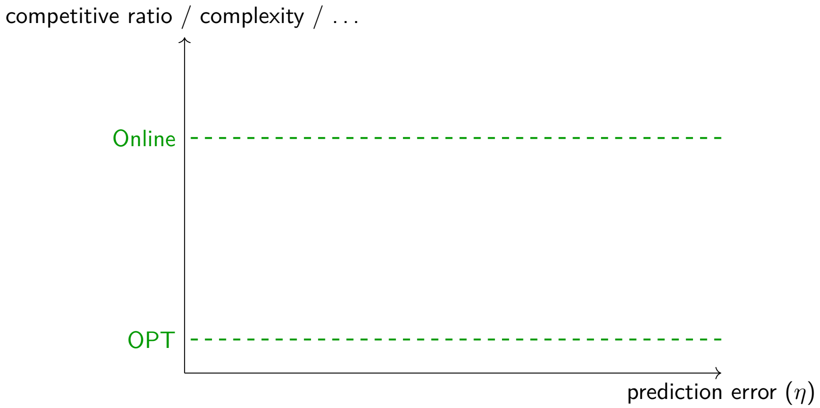


Prediction: position $h(q)$ Error: $\eta = |h(q) - index(q)|$

Classic: $\Theta(\log n)$ —predictions→ $\Theta(\log \eta)$

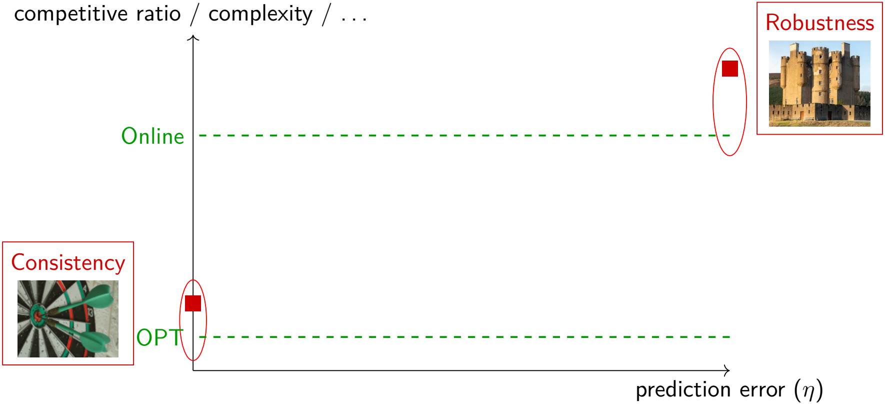
Practical applications [KraskaBCDP '18]

Properties we seek



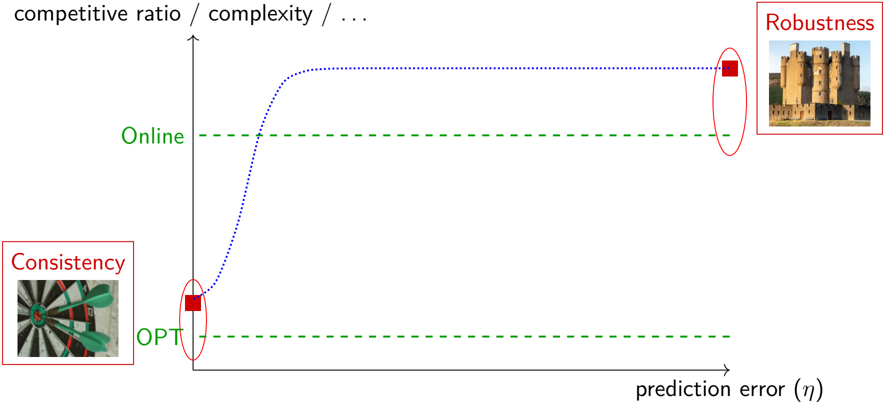
Algorithms are oblivious to η

Properties we seek



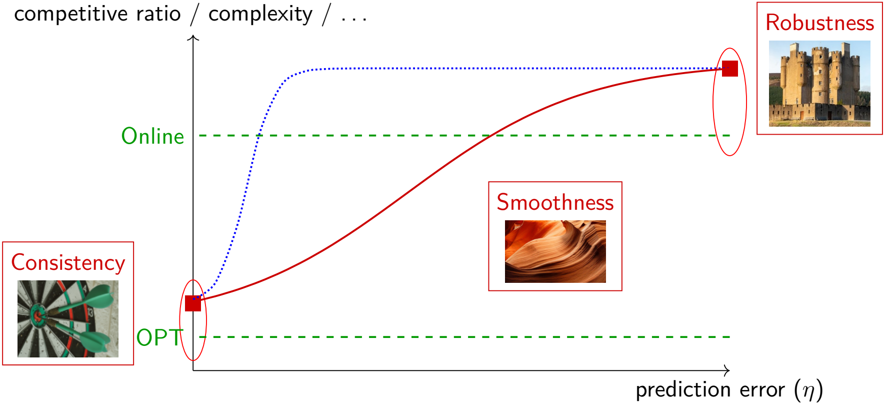
Algorithms are oblivious to η

Properties we seek



Algorithms are oblivious to η

Properties we seek



Algorithms are oblivious to η

Contract algorithms & contract scheduling

Contract algorithms

- Inputs include allowed processing time
- Performance improves if more time is allotted

Contract scheduling

- contract algorithm ⇒ *anytime* algorithm
  (anytime = can get interrupted "any time" and outputs a solution)

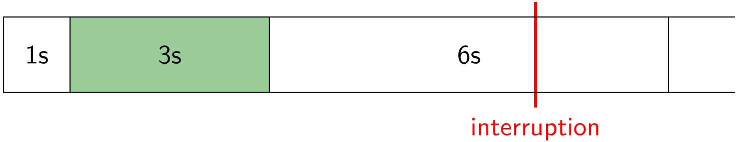
Contract algorithms & contract scheduling

Contract algorithms

- Inputs include allowed processing time
- Performance improves if more time is allotted

Contract scheduling

- contract algorithm ⇒ *anytime* algorithm
  (anytime = can get interrupted "any time" and outputs a solution)



Example: interruption at 8s, largest contract executed = 3s

Acceleration ratio

Schedule definition

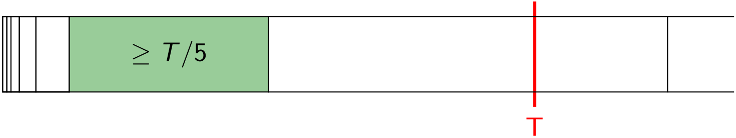
- sequence $\boldsymbol{X = \{x_i\}_{i \in \mathbb{Z}}}$, $x_i = i$th execution length
- start at $-\infty$ so that no interruption happens before 1st execution
- performance of X if interruption at T: $\boldsymbol{\ell(X, T)}$
  (length of the last contract terminated by X at time T)

Quality of a schedule

- acceleration ratio:

$$\mathrm{acc}(X) = \sup_T \frac{T}{\ell(X, T)}$$

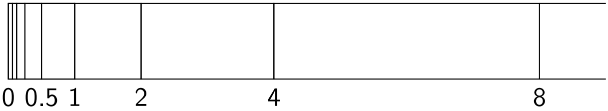
e.g., $\mathrm{acc}(X) = 5 \Rightarrow \forall T$, a contract has run for $\geq T/5$



Classic (no prediction) problem

Best contract algorithms: $X(\lambda)$ with $x_i = 2^{i+\lambda}$ for any $\lambda \in [0, 1]$

Example with $\lambda = 0$:



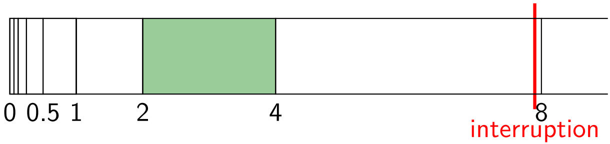
Proposition (from [Russel Zimberstein 91, Alpern Gal 03])

The acceleration ratio of $X(\lambda)$ is 4. All other algorithms are worse.

Classic (no prediction) problem

Best contract algorithms: $X(\lambda)$ with $x_i = 2^{i+\lambda}$ for any $\lambda \in [0,1]$

Example with $\lambda = 0$:



Proposition (from [Russel Zimberstein 91, Alpern Gal 03])

The acceleration ratio of $X(\lambda)$ is 4. All other algorithms are worse.

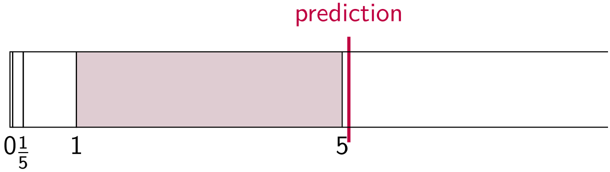
Previous work on single predictions [AK'23]

Framework

- prediction p = interruption time

Approach

- fix target robustness r, restrict to geometrical solutions.
  Candidate schedules are $\{a^{i+\lambda}\}_{i\in\mathbb{Z}}$ with $a \in \left[\frac{1}{2}(r \pm \sqrt{r^2-4r})\right]$
- best solution: largest a, shift λ to "aim" at p
- issue: not smooth
  explore prediction with bounded error



Outline

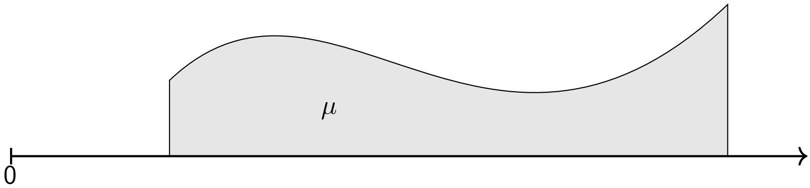
Distributional predictions

Framework

- prediction $\mu =$ prob. distr. of the interruption time
- to simplify: aim at best consistency while staying 4-robust
  only choice: $\lambda \in [0, 1]$ in $\{2^{i+\lambda}\}_{i\in\mathbb{Z}}$

Consistency definition

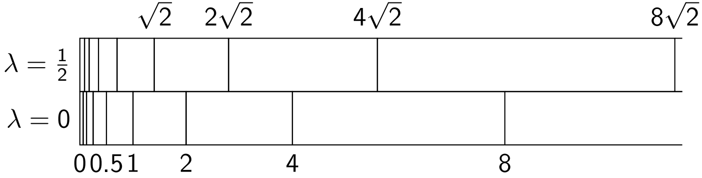
$$c(X, \mu) = \frac{\mathbb{E}_{T\sim\mu}[T]}{\mathbb{E}_{T\sim\mu}[\ell(X, T)]}$$



Can we do something without any assumption on μ ?

First idea

- take two opposite shifted schedules $\{2^{i+\lambda}\}_{i\in\mathbb{Z}}$: $(\lambda \in \{0, \frac{1}{2}\})$, select "the best"
- $\Rightarrow 8(\sqrt{2}-1)$ - consistent



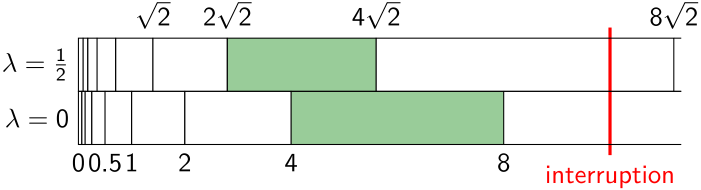
Best of n regularly shifted schedules

- $\Rightarrow 4n(2^{-1/n}-1)$ - consistent
- $\longrightarrow_{n\to\infty} 4\ln 2$

Can we do something without any assumption on μ ?

First idea

- take two opposite shifted schedules $\{2^{i+\lambda}\}_{i\in\mathbb{Z}} : (\lambda \in \{0, \frac{1}{2}\})$, select "the best"
- $\Rightarrow 8(\sqrt{2}-1)$ - consistent



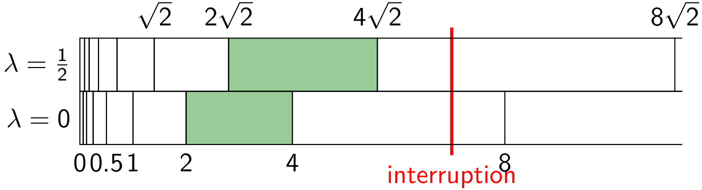
Best of n regularly shifted schedules

- $\Rightarrow 4n(2^{-1/n}-1)$ - consistent
- $\longrightarrow_{n\to\infty} 4\ln 2$

Can we do something without any assumption on μ ?

First idea

- take two opposite shifted schedules $\{2^{i+\lambda}\}_{i\in\mathbb{Z}}$: ($\lambda \in \{0, \frac{1}{2}\}$), select "the best"
- $\Rightarrow 8(\sqrt{2}-1)$ - consistent



Best of n regularly shifted schedules

- $\Rightarrow 4n(2^{-1/n}-1)$ - consistent
- $\longrightarrow_{n\to\infty} 4\ln 2$

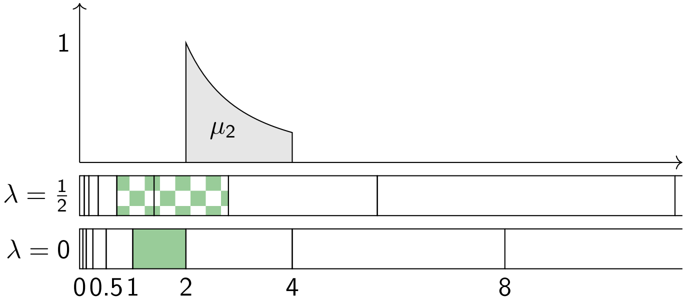
Distributional lower bound

Summary

- the previous solutions are tight

Theorem

For any D and μ_D having a density $f_D(x) = \frac{2D}{x^2}$ on $[D; 2D]$, no 4-robust schedule has a consistency better than $4 \ln 2$.



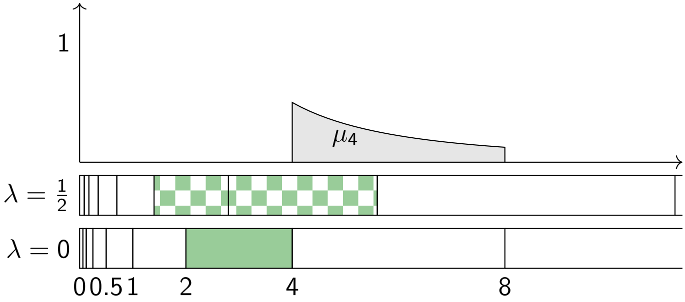
Distributional lower bound

Summary

- the previous solutions are tight

Theorem

For any D and μ_D having a density $f_D(x) = \frac{2D}{x^2}$ on $[D;2D]$, no 4-robust schedule has a consistency better than $4\ln 2$.



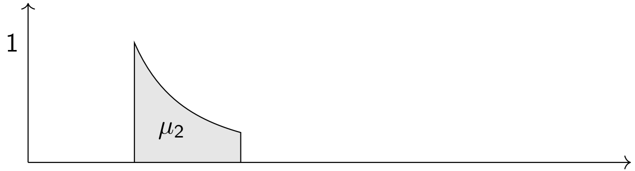
"Smoothness"

Recall on single predictions

- best algorithm: ε error on prediction destroys everything

On distributed predictions

- intuitively: worst-case predictions needs *balanced* probability mass
  small perturbation $\Rightarrow$ small impact on performance
  same worst-case for all 4-robust schedules
- formally: perturbation measured via Earth-Mover Distance
- technical result: if the actual distribution is close to μ_D (wrt EMD), then the acceleration ratio of any 4-robust schedule is close to $4 \ln 2$



Outline

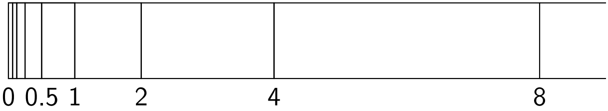
Prediction = multiple advice

Framework

- prediction $P = \{\tau_1, \ldots, \tau_k\}$
- goal: optimize performance wrt adversarial interruption among P

Consistency definition

- $c(X, P) = \sup_{\tau \in P} \frac{\tau}{\ell(X, \tau)}$



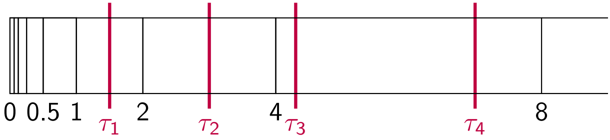
Prediction = multiple advice

Framework

- prediction $P = \{\tau_1, \ldots, \tau_k\}$
- goal: optimize performance wrt adversarial interruption among P

Consistency definition

- $c(X, P) = \sup_{\tau \in P} \frac{\tau}{\ell(X, \tau)}$



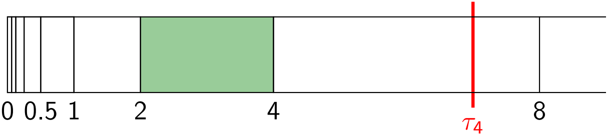
Prediction = multiple advice

Framework

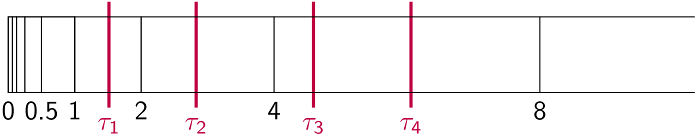
- prediction $P = \{\tau_1, \ldots, \tau_k\}$
- goal: optimize performance wrt adversarial interruption among P

Consistency definition

- $c(X, P) = \sup_{\tau \in P} \frac{\tau}{\ell(X, \tau)}$



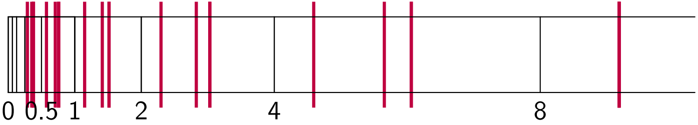
Algorithm



$\tau_j = 2^{i_j+\lambda_j}$, such that $i_j \in \mathbb{Z}, \lambda_j \in [0, 1]$

$$ex : \{\lambda_j\} = \{0.6, 0.5, 0.2, 0.6\}$$

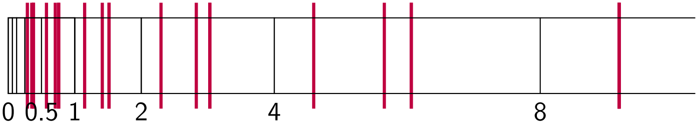
Algorithm



$$\tau_j = 2^{i_j+\lambda_j}, \text{ such that } i_j \in \mathbb{Z}, \lambda_j \in [0,1] \;\approx \tau_j = \{2^{i+\lambda_j}\}_{i\in\mathbb{Z}}$$

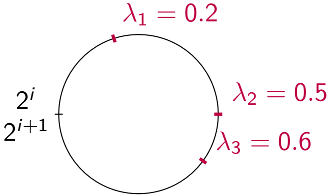
$$ex : \{\lambda_j\} = \{0.6, 0.5, 0.2, 0.6\}$$

Algorithm

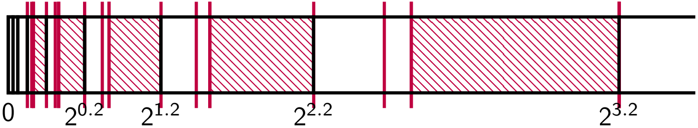


$$\tau_j = 2^{i_j+\lambda_j}, \text{ such that } i_j \in \mathbb{Z}, \lambda_j \in [0,1] \;\approx \tau_j = \{2^{i+\lambda_j}\}_{i\in\mathbb{Z}}$$

$$ex : \{\lambda_j\} = \{0.6, 0.5, 0.2, 0.6\}$$

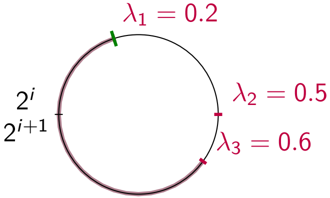


Algorithm



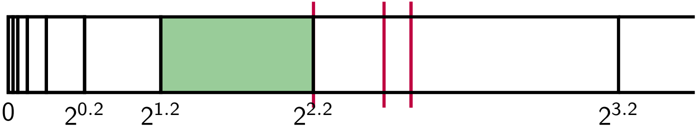
$$\tau_j = 2^{i_j+\lambda_j}, \text{ such that } i_j \in \mathbb{Z}, \lambda_j \in [0,1] \ \approx \tau_j = \{2^{i+\lambda_j}\}_{i\in\mathbb{Z}}$$

$$ex : \{\lambda_j\} = \{0.6, 0.5, 0.2, 0.6\}$$



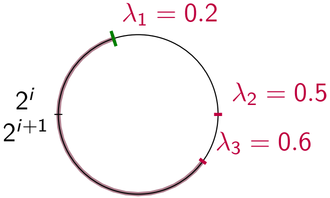
$$\Delta = \max_i(\lambda_{1+i \bmod k} - \lambda_i)$$

Algorithm



$$\tau_j = 2^{i_j+\lambda_j}, \text{ such that } i_j \in \mathbb{Z}, \lambda_j \in [0,1] \;\; \approx \tau_j = \{2^{i+\lambda_j}\}_{i\in\mathbb{Z}}$$

$$ex : \{\lambda_j\} = \{0.6, 0.5, 0.2, 0.6\}$$

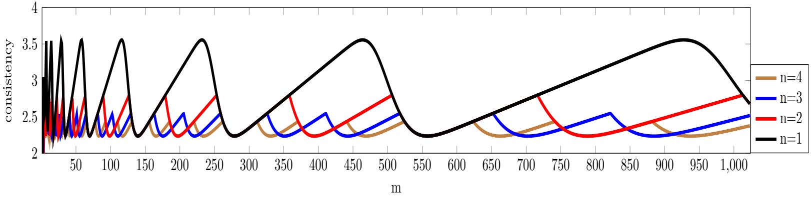


$$\Delta = \max_i(\lambda_{1+i \bmod k} - \lambda_i)$$

Consistency of choosing λ "targeting" Δ: $2^{2-\Delta} \geq 2^{2-\frac{1}{k}}$ (this is tight)

Outline

Distributional predictions



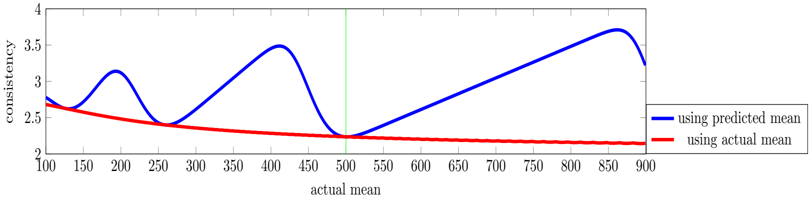
Setup

- Algo: choose best among n schedules $\{2^{i+k/n}\}_{i\in\mathbb{Z}}$ for $k \in [1 \ldots n]$
- Prediction: truncated normal distribution mean m st. dev. $0.05m$
- Plot consistency in function of m (bottom is best)

Remarks

- Larger n = minimum of more functions
- Steeper downward slope (worse to interrupt before a contract)

Distributional predictions : “smoothness”



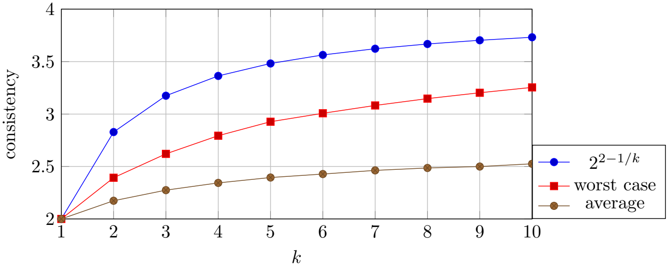
Setup

- Algo: choose best among 16 schedules $\{2^{i+k/n}\}_{i\in\mathbb{Z}}$ for $k \in [1 \ldots 16]$
- Prediction: - Top curve: truncated normal distribution mean 500, $\sigma = 25$
  - Bottom curve: truncated normal distribution mean m, $\sigma = 25$
- Interruption: truncated normal distribution mean m, $\sigma = 25$
- Plot ratio m over the expected performance

Remarks

- Smooth asymetric degradation with the error (linked to σ)

Multiple predictions



Setup

- Prediction P: $k \in [1 \dots 10]$ candidate times drawn $\mathcal{U}(1, 1024)$
- Plot: - theoretical consistency
  - experimental consistency, averaged over 1000 repetitions
  - experimental perf. if interruption drawn uniformly from P

Remarks

- Results with non-pathological predictions much better than theoretical bounds

Conclusion

Framework

- objective: study models beyond simple prediction
- original idea: prediction as probability distribution

Results

- simple algorithms best consistency when robustness $= 4$
- hard to get more general results

Future direction

- focus on a simpler related problem to aim at more general results: online bidding