# Online Computation with Untrusted Advice

Adrian Siwiec

Jagiellonian University

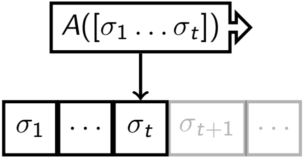
April 16, 2020

# References

**Spyros Angelopoulos, Christoph Dürr, Shendan Jin, Shahin Kamali, Marc Renault**
*"Online Computation with Untrusted Advice"*
https://arxiv.org/pdf/1905.05655.pdf

# Online Algorithms

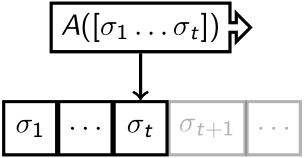
## Online Algorithm




### Total Cost

$$A(\sigma) := \sum_t A([\sigma_1 \dots \sigma_t])$$

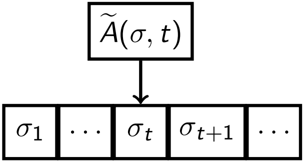
## Online Algorithm




**Total Cost**

$A(\sigma) := \sum_t A([\sigma_1 \dots \sigma_t])$

## Offline Algorithm

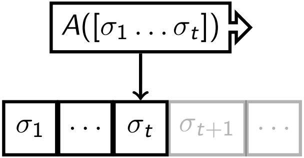



**Total Cost**

Cost: $\widetilde{A}(\sigma) := \sum_t \widetilde{A}(\sigma, t)$

# Online Algorithms

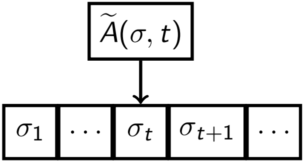
## Online Algorithm




**Total Cost**

$$A(\sigma) := \sum_t A([\sigma_1 \ldots \sigma_t])$$

## Offline Algorithm



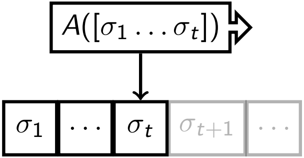

**Total Cost**

Cost: $\widetilde{A}(\sigma) := \sum_t \widetilde{A}(\sigma, t)$

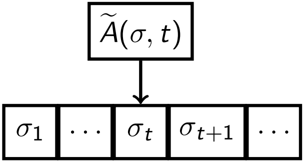
$$OPT(\sigma) := \inf_{\widetilde{A}} \widetilde{A}(\sigma)$$

# Online Algorithms

**Online Algorithm**




**Offline Algorithm**




**Total Cost**

$$A(\sigma) := \sum_t A([\sigma_1 \ldots \sigma_t])$$

**Total Cost**

Cost: $\widetilde{A}(\sigma) := \sum_t \widetilde{A}(\sigma, t)$

$$OPT(\sigma) := \inf_{\widetilde{A}} \widetilde{A}(\sigma)$$

**Competitive Ratio**

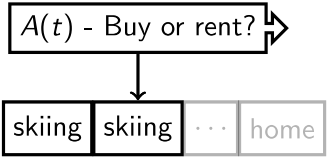
$$CR(A) = \sup_\sigma \frac{A(\sigma)}{OPT(\sigma)}$$

# Ski Rental Problem

## Ski Rental Problem

You are going skiing for an unknown number of days. (Denoted $D$)
Renting skis: \$1 per day.
Buying skis: $\$B$.

**Online Algorithm**



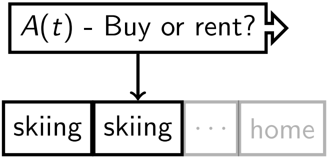
## Ski Rental Problem

You are going skiing for an unknown number of days. (Denoted $D$)
Renting skis: \$1 per day.
Buying skis: \$$B$.

## OPT

$\min D, B$

**Online Algorithm**



# Ski Rental Problem

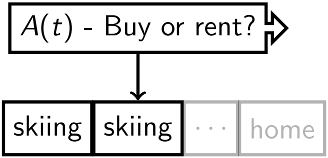
## Ski Rental Problem

You are going skiing for an unknown number of days. (Denoted $D$)
Renting skis: \$1 per day.
Buying skis: $\$B$.

## OPT

$\min D, B$

**Online Algorithm**

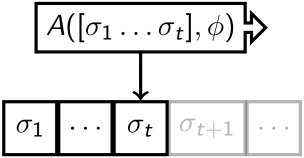



## Break-Even

Rent $B-1$ days. Buy on day $B$ if still skiing.
$CR(\text{Break-Even}) = \frac{(B-1)+B}{B} \approx 2$

# Online algorithms with advice

**Online Algorithm**

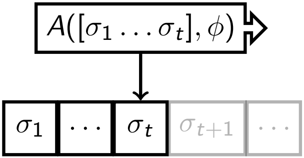



**Advice is some offline information**

$$A(\sigma, \phi) := \sum_t A([\sigma_1 \dots \sigma_t], \phi)$$

# Online algorithms with advice

**Online Algorithm**




## Advice is some offline information

$$A(\sigma, \phi) := \sum_t A([\sigma_1 \ldots \sigma_t], \phi)$$

## Advice complexity

Algorithm $A$ has *advice complexity* $s(n)$ if for every request $\sigma$ of length $n$ it uses only the first $s(n)$ bits of advice $\phi$.

# Back to our example

## Ski Rental Problem

$\phi \in \{0, 1\}$ – Rent everyday or buy on day 1.
$CR(A) = 1$
$s(n) = 1$

# Back to our example

## Ski Rental Problem

$\phi \in \{0, 1\}$ – Rent everyday or buy on day 1.
$CR(A) = 1$
$s(n) = 1$

## Ski Rental – Corrupted or adversary advice.

Advice is to rent, number of days is very big.
$CR(A) \to \infty$

# Untrusted advice

## Consistency

$$r_A = \sup_{\sigma} \inf_{\phi} \frac{A(\sigma,\phi)}{OPT(\sigma)}$$

## Robustness

$$w_A = \sup_{\sigma} \sup_{\phi} \frac{A(\sigma,\phi)}{OPT(\sigma)}$$

# Untrusted advice

## Consistency

$$r_A = \sup_{\sigma} \inf_{\phi} \frac{A(\sigma,\phi)}{OPT(\sigma)}$$

## Robustness

$$w_A = \sup_{\sigma} \sup_{\phi} \frac{A(\sigma,\phi)}{OPT(\sigma)}$$

Our trusting ski rental algorithm has consistency of 1, but unbounded robustness.
We will denote it as $(1, \infty)$-competetive.

# Untrusted advice

## Consistency

$$r_A = \sup_{\sigma} \inf_{\phi} \frac{A(\sigma,\phi)}{OPT(\sigma)}$$

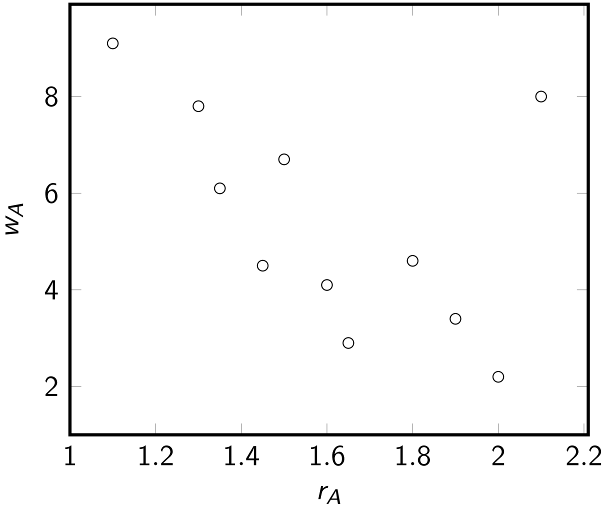
## Robustness

$$w_A = \sup_{\sigma} \sup_{\phi} \frac{A(\sigma,\phi)}{OPT(\sigma)}$$

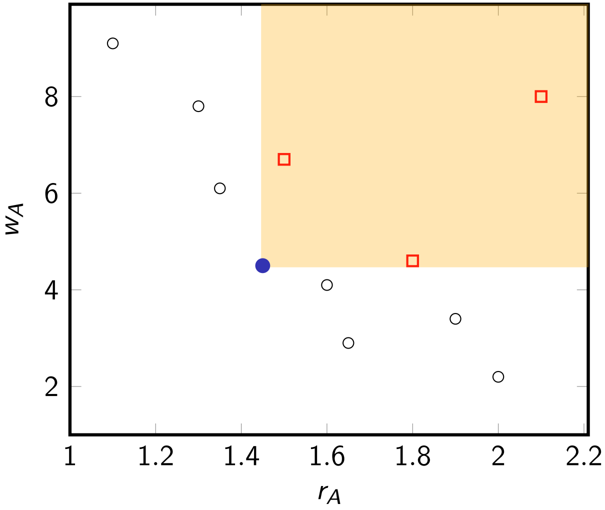
Our trusting ski rental algorithm has consistency of 1, but unbounded robustness.
We will denote it as $(1, \infty)$-competetive.

An algorithm ignoring advice is $(2, 2)$-competetive.

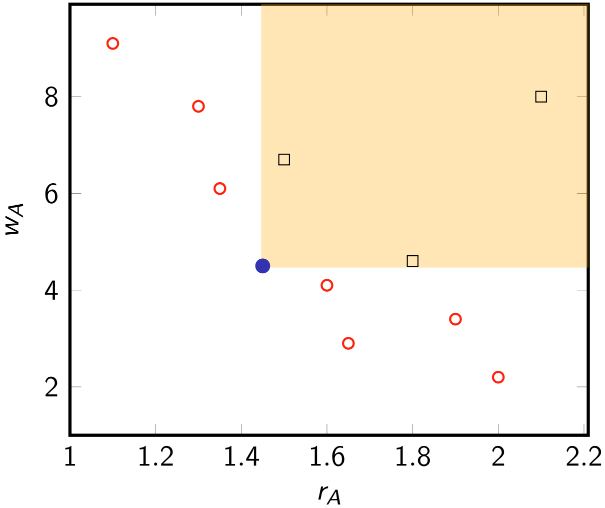
# Pareto efficiency



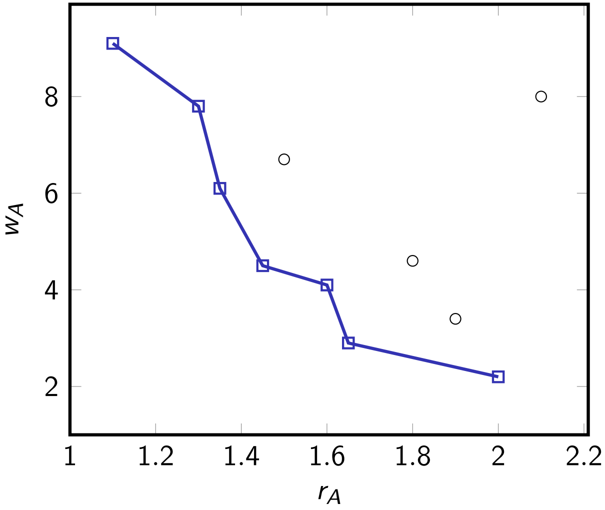
# Pareto dominance



# Pareto dominance



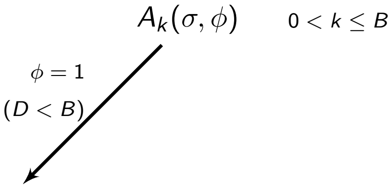
# Pareto frontier



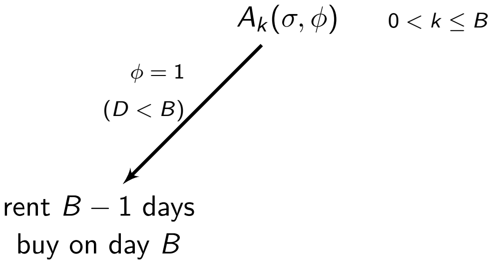
# Back to ski rental

$$A_k(\sigma, \phi) \qquad 0 < k \le B$$

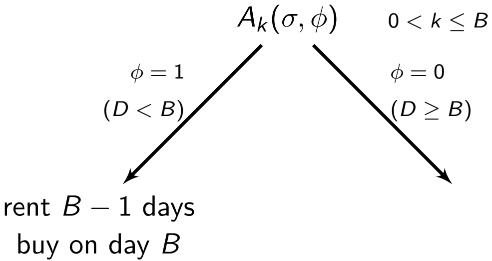
# Back to ski rental



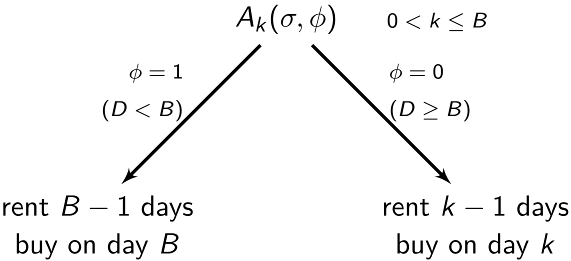
# Back to ski rental



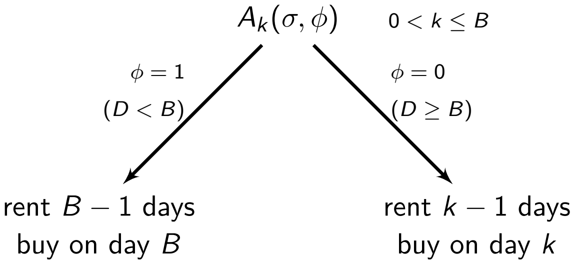
# Back to ski rental



# Back to ski rental



# Back to ski rental




**Algorithm $A_k$**

Algorithm $A_k$ is $(1 + \frac{k-1}{B}, 1 + \frac{B-1}{k})$-competetive.

# Ski rental competetive ratio

## Algorithm $A_k$

Algorithm $A_k$ is $(1 + \frac{k-1}{B}, 1 + \frac{B-1}{k})$-competetive.

# Ski rental competetive ratio

## Algorithm $A_k$

Algorithm $A_k$ is $(1+\frac{k-1}{B}, 1+\frac{B-1}{k})$-competetive.

| advice | trusted | untrusted |
|---|---|---|
| 0 $(D \geq B)$ | $\frac{k-1+B}{B} = 1+\frac{k-1}{B}$ | $\frac{k-1+B}{k} = 1+\frac{B-1}{k}$ |
| 1 $(D < B)$ | 1 | $2-\frac{1}{B}$ |

# Ski rental competetive ratio

## Algorithm $A_k$

Algorithm $A_k$ is $(1 + \frac{k-1}{B}, 1 + \frac{B-1}{k})$-competetive.

| advice | trusted | untrusted |
|---|---|---|
| $0\ (D \geq B)$ | $\frac{k-1+B}{B} = 1 + \frac{k-1}{B}$ | $\frac{k-1+B}{k} = 1 + \frac{B-1}{k}$ |
| $1\ (D < B)$ | $1$ | $2 - \frac{1}{B}$ |

## Theorem

*Algorithms $A_k$ are Pareto-optimal.*

# $A_k$ is Pareto-optimal

**Theorem**

*For any deterministic* $(1 + \frac{k-1}{B}, w)$*-competetive algorithm* $A$*, with* $1 \leq k \leq B$ *and advice of any size, it holds that* $w \geq 1 + \frac{B-1}{k}$*.*

Let $A$ have trusted competetive ratio at most $1 + \frac{k-1}{B}$.

# $A_k$ is Pareto-optimal

## Theorem

*For any deterministic* $(1 + \frac{k-1}{B}, w)$*-competetive algorithm* $A$*, with* $1 \leq k \leq B$ *and advice of any size, it holds that* $w \geq 1 + \frac{B-1}{k}$.

Let $A$ have trusted competetive ratio at most $1 + \frac{k-1}{B}$.

- $k = B$

# $A_k$ is Pareto-optimal

**Theorem**

*For any deterministic $(1 + \frac{k-1}{B}, w)$-competetive algorithm $A$, with $1 \leq k \leq B$ and advice of any size, it holds that $w \geq 1 + \frac{B-1}{k}$.*

Let $A$ have trusted competetive ratio at most $1 + \frac{k-1}{B}$.

- $k = B$
- Algorithm with untrusted advice can not be better than purely online algorithm.

# $A_k$ is Pareto-optimal

**Theorem**

*For any deterministic $(1 + \frac{k-1}{B}, w)$-competetive algorithm $A$, with $1 \leq k \leq B$ and advice of any size, it holds that $w \geq 1 + \frac{B-1}{k}$.*

Let $A$ have trusted competetive ratio at most $1 + \frac{k-1}{B}$.

- $k = B$
- Algorithm with untrusted advice can not be better than purely online algorithm.
- Best possible competitive ratio for purely online algorithm is $1 + \frac{B-1}{B}$.

# $A_k$ is Pareto-optimal

**Theorem**

*For any deterministic $(1 + \frac{k-1}{B}, w)$-competetive algorithm A, with $1 \leq k \leq B$ and advice of any size, it holds that $w \geq 1 + \frac{B-1}{k}$.*

Let $A$ have trusted competetive ratio at most $1 + \frac{k-1}{B}$.

- $k < B$

# $A_k$ is Pareto-optimal

**Theorem**

*For any deterministic $(1 + \frac{k-1}{B}, w)$-competetive algorithm $A$, with $1 \leq k \leq B$ and advice of any size, it holds that $w \geq 1 + \frac{B-1}{k}$.*

Let $A$ have trusted competetive ratio at most $1 + \frac{k-1}{B}$.

- $k < B$
- If advice is trusted, for input $\sigma_{B+k}$ we have: $A(\sigma_{B+k}) \leq (1 + \frac{k-1}{B}) OPT(\sigma_{B+k})$

# $A_k$ is Pareto-optimal

## Theorem

*For any deterministic* $(1 + \frac{k-1}{B}, w)$*-competetive algorithm* $A$*, with* $1 \leq k \leq B$ *and advice of any size, it holds that* $w \geq 1 + \frac{B-1}{k}$*.*

Let $A$ have trusted competetive ratio at most $1 + \frac{k-1}{B}$.

- $k < B$
- If advice is trusted, for input $\sigma_{B+k}$ we have: $A(\sigma_{B+k}) \leq B + k - 1$

# $A_k$ is Pareto-optimal

**Theorem**

*For any deterministic $(1 + \frac{k-1}{B}, w)$-competetive algorithm $A$, with $1 \leq k \leq B$ and advice of any size, it holds that $w \geq 1 + \frac{B-1}{k}$.*

Let $A$ have trusted competetive ratio at most $1 + \frac{k-1}{B}$.

- $k < B$
- If advice is trusted, for input $\sigma_{B+k}$ we have: $A(\sigma_{B+k}) \leq B + k - 1$
- Let us assume $A(\sigma_{B+k}, x)$ buys on day $j \leq k$ with trusted advice $x$.

# $A_k$ is Pareto-optimal

**Theorem**

*For any deterministic $(1 + \frac{k-1}{B}, w)$-competetive algorithm $A$, with $1 \leq k \leq B$ and advice of any size, it holds that $w \geq 1 + \frac{B-1}{k}$.*

Let $A$ have trusted competetive ratio at most $1 + \frac{k-1}{B}$.

- $k < B$
- If advice is trusted, for input $\sigma_{B+k}$ we have: $A(\sigma_{B+k}) \leq B + k - 1$
- Let us assume $A(\sigma_{B+k}, x)$ buys on day $j \leq k$ with trusted advice $x$.
- $A(\sigma_j, x)$ will do the same – $x$ here will be untrusted.

# $A_k$ is Pareto-optimal

**Theorem**

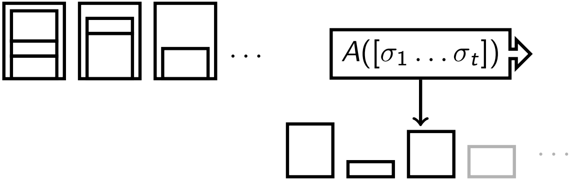
*For any deterministic $(1 + \frac{k-1}{B}, w)$-competetive algorithm $A$, with $1 \leq k \leq B$ and advice of any size, it holds that $w \geq 1 + \frac{B-1}{k}$.*

Let $A$ have trusted competetive ratio at most $1 + \frac{k-1}{B}$.

- $k < B$
- If advice is trusted, for input $\sigma_{B+k}$ we have: $A(\sigma_{B+k}) \leq B + k - 1$
- Let us assume $A(\sigma_{B+k}, x)$ buys on day $j \leq k$ with trusted advice $x$.
- $A(\sigma_j, x)$ will do the same – $x$ here will be untrusted.
- After some simple calculations we get $w_{A_k} \geq 1 + \frac{B-1}{k}$

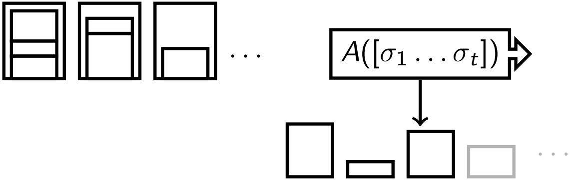
# And Now for Something Completely Different

**Bin Packing**



# And Now for Something Completely Different

**Bin Packing**




### Competetive ratio

$CR(A) = \min_r : \forall\sigma : A(\sigma) \leq r * OPT(\sigma) + c$

# Known algorithms

## First fit

$CR(\textit{first-fit}) = 1.7$

# Known algorithms

## First fit

$CR(\textit{first-fit}) = 1.7$

## Best Online

$1.54278 \leq CR(\textit{best-online}) \leq 1.5783$

# Known algorithms

**First fit**

$CR(\textit{first-fit}) = 1.7$

**Best Online**

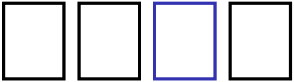
$1.54278 \leq CR(\textit{best-online}) \leq 1.5783$

**Best constant-size trusted advice**

$CR(\textit{best-trusted}) = 1.4702$

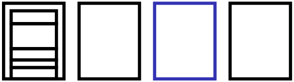
# Reserve-Critical

**Reserve-Critical**($c$)




$(0, \frac{1}{3}]$ $(\frac{1}{3}, \frac{1}{2}]$ $(\frac{1}{2}, \frac{2}{3}]$ $(\frac{2}{3}, 1]$

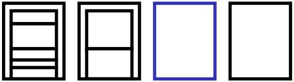
**Reserve-Critical**($c$)



| tiny | small | critical | large |
|---|---|---|---|
| $(0, \frac{1}{3}]$ | $(\frac{1}{3}, \frac{1}{2}]$ | $(\frac{1}{2}, \frac{2}{3}]$ | $(\frac{2}{3}, 1]$ |

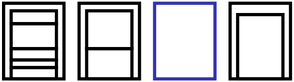
# Reserve-Critical

**Reserve-Critical**($c$)



tiny $(0, \frac{1}{3}]$ | small $(\frac{1}{3}, \frac{1}{2}]$ | critical $(\frac{1}{2}, \frac{2}{3}]$ | large $(\frac{2}{3}, 1]$

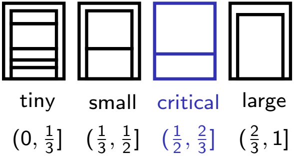
**Reserve-Critical($c$)**



| tiny | small | critical | large |
| --- | --- | --- | --- |
| $(0, \frac{1}{3}]$ | $(\frac{1}{3}, \frac{1}{2}]$ | $(\frac{1}{2}, \frac{2}{3}]$ | $(\frac{2}{3}, 1]$ |

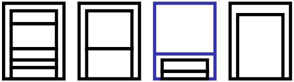
# Reserve-Critical

**Reserve-Critical**($c$)



# Reserve-Critical

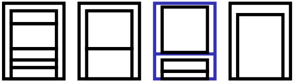
**Reserve-Critical**($c$)



tiny | small | critical | large

$(0, \frac{1}{3}]$ | $(\frac{1}{3}, \frac{1}{2}]$ | $(\frac{1}{2}, \frac{2}{3}]$ | $(\frac{2}{3}, 1]$

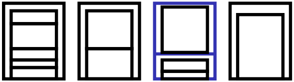
**Reserve-Critical**($c$)



| tiny | small | critical | large |
|---|---|---|---|
| $(0, \frac{1}{3}]$ | $(\frac{1}{3}, \frac{1}{2}]$ | $(\frac{1}{2}, \frac{2}{3}]$ | $(\frac{2}{3}, 1]$ |

# Reserve-Critical

**Reserve-Critical**($c$)

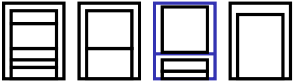


| tiny | small | critical | large |
|---|---|---|---|
| $(0, \frac{1}{3}]$ | $(\frac{1}{3}, \frac{1}{2}]$ | $(\frac{1}{2}, \frac{2}{3}]$ | $(\frac{2}{3}, 1]$ |

What if advice $c$ is untrusted?

# Robust-Reserve-Critical

**Robust-Reserve-Critical**$(\gamma, \alpha \in [0,1])$

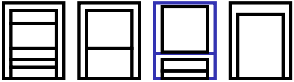


| tiny | small | critical | large |
|---|---|---|---|
| $(0, \frac{1}{3}]$ | $(\frac{1}{3}, \frac{1}{2}]$ | $(\frac{1}{2}, \frac{2}{3}]$ | $(\frac{2}{3}, 1]$ |

*Critical ratio* $= \frac{c}{c+t}$

# Robust-Reserve-Critical

**Robust-Reserve-Critical**($\gamma, \alpha \in [0,1]$)

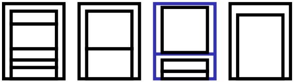


| tiny | small | critical | large |
|---|---|---|---|
| $(0, \frac{1}{3}]$ | $(\frac{1}{3}, \frac{1}{2}]$ | $(\frac{1}{2}, \frac{2}{3}]$ | $(\frac{2}{3}, 1]$ |

*Critical ratio* $= \frac{c}{c+t}$
*RRC* maintains critical ratio of $\beta = min\{\gamma, \alpha\}$

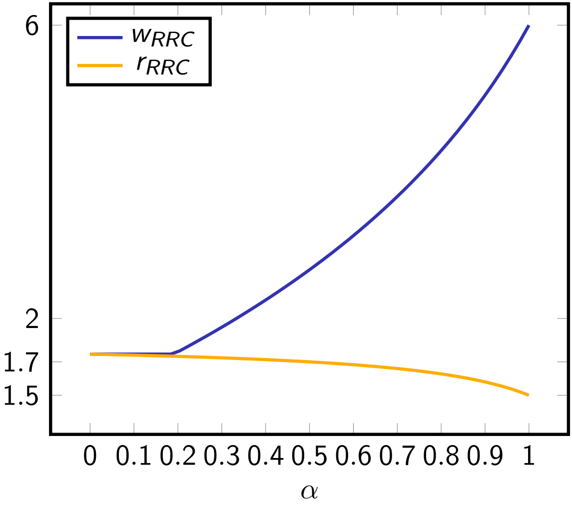
# Robust-Reserve-Critical

**Robust-Reserve-Critical**$(\gamma, \alpha \in [0,1])$

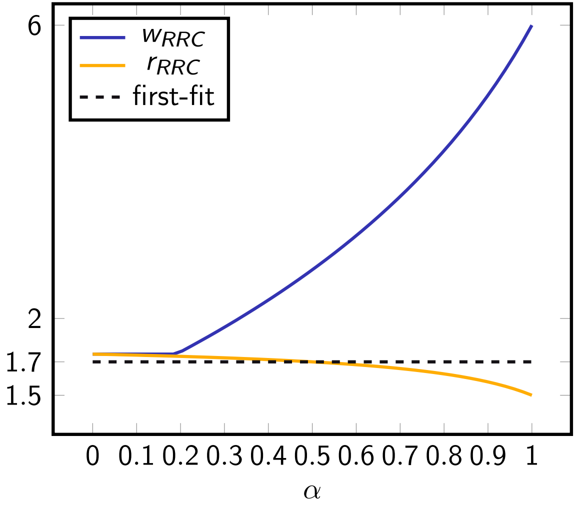



*Critical ratio* $= \frac{c}{c+t}$
*RRC* maintains critical ratio of $\beta = min\{\gamma, \alpha\}$
*RRC* is $(1.5 + \frac{1-\alpha}{4-3\alpha}, 1.5 + max\{\frac{1}{4}, \frac{9\alpha}{8-6\alpha}\})$-competetive

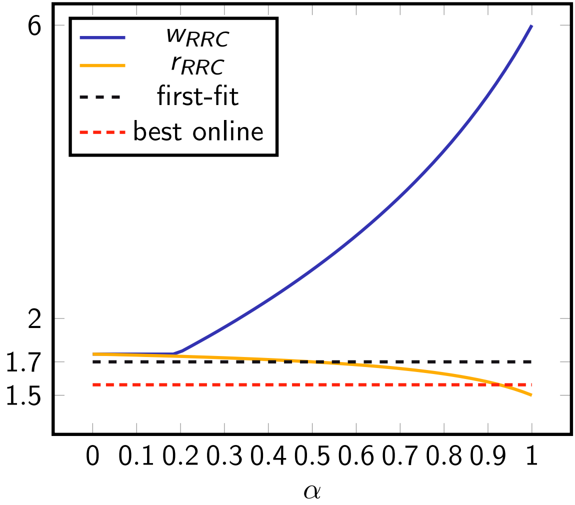
# Bin packing algorithms



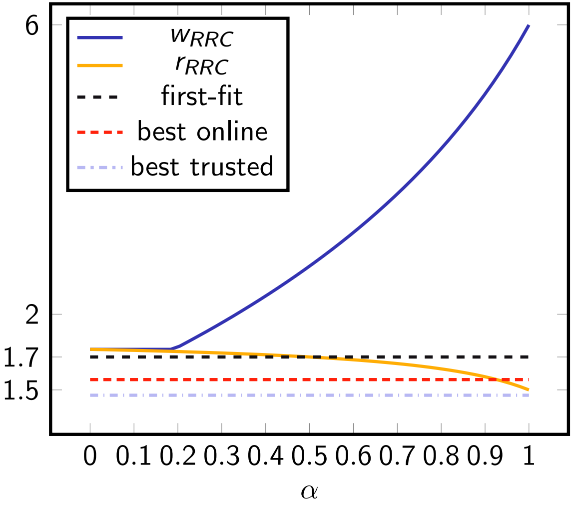
# Bin packing algorithms



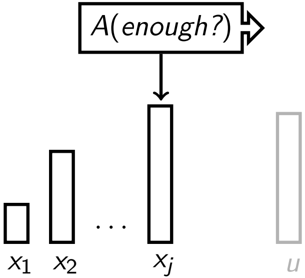
# Bin packing algorithms



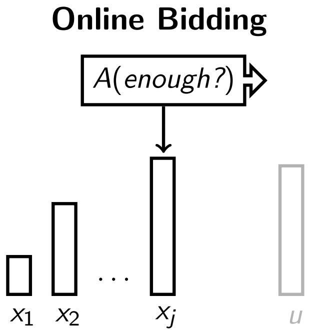
# Bin packing algorithms



## Online Bidding



# Online Bidding




## Competetive Ratio

$w_A = \sup_u \frac{\sum_{i=1}^{j} x_i}{u}$, where $j$ is such that $x_{j-1} < u \le x_j$.

# Online Bidding – Algorithms

## Doubling algorithm

Best algorithm without advice is a doubling algorithm. It is $(4, 4)$-competetive.

## Trusting algorithm

Algorithm that completely trusts advice $u$ is $(1, \infty)$-competetive.

## Pareto-Optimal algorithm

Pareto-Optimal algorithm is $(\frac{w-\sqrt{w^2-4w}}{2}, w)$-competetive.

# Pareto-Optimal Online Bidding – Overwiew

- Define class $S_{m,u}$, $m \in \mathbb{N}^+$ as the set of bidding strategies with advice $u$, that are $w$-competetive, and succedd on $m$-th bid with trusted advice.

# Pareto-Optimal Online Bidding – Overwiew

- Define class $S_{m,u}$, $m \in \mathbb{N}^+$ as the set of bidding strategies with advice $u$, that are $w$-competetive, and succedd on $m$-th bid with trusted advice.
- Let $X^*_{m,u}$ be a dominant strategy in $S_{m,u}$, which is $(r_{m,u}, w)$-competetive.

# Pareto-Optimal Online Bidding – Overwiew

- Define class $S_{m,u}$, $m \in \mathbb{N}^+$ as the set of bidding strategies with advice $u$, that are $w$-competetive, and succedd on $m$-th bid with trusted advice.
- Let $X^*_{m,u}$ be a dominant strategy in $S_{m,u}$, which is $(r_{m,u}, w)$-competetive.
- $X^*_{m,u}$ is a solution of an infinite linear program:

$$\begin{aligned}
\min \quad & r_{m,u} && (P_{m,u}) \\
\text{s.t.} \quad & x_i < x_{i+1}, \quad i \in \mathbb{N}^+ \\
& x_{m-1} < u \le x_m \\
& \sum_{j=1}^{m} x_j \le r_{m,u} \cdot u \\
& \sum_{j=1}^{i} x_j \le w \cdot x_{i-1}, \quad i \in \mathbb{N}^+ \\
& x_i \ge 0, \quad i \in \mathbb{N}^+.
\end{aligned}$$

# Online Bidding