Scheduling Jobs on Grid Processors

Joan Boyar

Lene M. Favrholdt

Department of Mathematics and Computer Science

University of Southern Denmark, Odense

The Grid

Grid computing:

- wide area distributed computing
- “A New Infrastructure for 21st Century Science”
- built on the Internet
- analogous to electical power grid
  - source and location of processors invisible
  - request resources (processors with memory)
  - pay for resources used

Grid Scheduling Problem

- Jobs: $J_1, J_2, \ldots, J_n$ given initially
  job J_i has requirement p_i
- Processors: $P_1, P_2, \ldots, P_k$ arrive online
  processor P_j has capacity c_j
- Goal: Minimize total capacity of processors used

Grid Scheduling Problem

- Jobs: J_1, J_2,..., J_n given initially
  job J_i has requirement p_i
- Processors: P_1, P_2,..., P_k arrive online
  processor P_j has capacity c_j
- Goal: Minimize total capacity of processors used

Bin Packing Problem [G. Zhang '97]

- Items: sizes $\in \{1, 2, ..., B\}$: s_1, s_2,..., s_n
- Bins: sizes $\in \{1, 2, ..., B\}$: b_1, b_2,..., b_k
  - arrive on-line
  - pack current bin before next arrives
- Goal: Minimize total size of bins used
- Restriction: Must use bin if any remaining item fits

Competitive Ratio

$\mathbb{A}$ is c-competitive if for any input seq. I,

$$\mathbb{A}(I) \leq c \cdot \mathsf{OPT}(I) + b.$$

optimal off-line algorithm (OPT) — constant (b)

The competitive ratio of $\mathbb{A}$ is

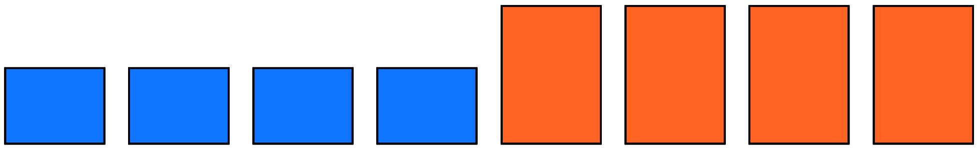
$$\mathsf{CR}_{\mathbb{A}} = \inf \{ c \mid \mathbb{A} \text{ is } c\text{-competitive} \}.$$

Grid Scheduling Algorithms

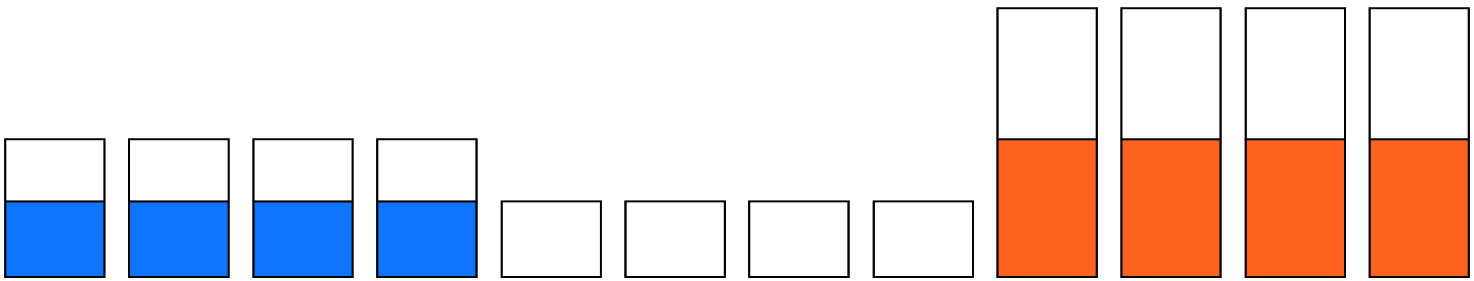
- FFI — First-Fit Increasing
- FFD — First-Fit Decreasing
  - searches entire list of items
- FFD$_\alpha$ $(1/2 < \alpha \leq 1)$
  - try FFD for each item size $B, B-1, ..., 1$
  - stop looking if bin filled to $\geq \alpha$
    - $\alpha \leq 1/2$: FFD$_\alpha$ same as FFD
    - $\alpha < 3/4$: FFD$_\alpha$ “same” as FFD on identical bins
    - $\alpha > 3/4$: can be worse than FFD on identical bins

FFI — First-Fit Increasing

$B = 40$.
Item sizes: $4 \times [11], 4 \times [20]$
Bin sizes: $4 \times [20], 4 \times [11], 4 \times [39]$



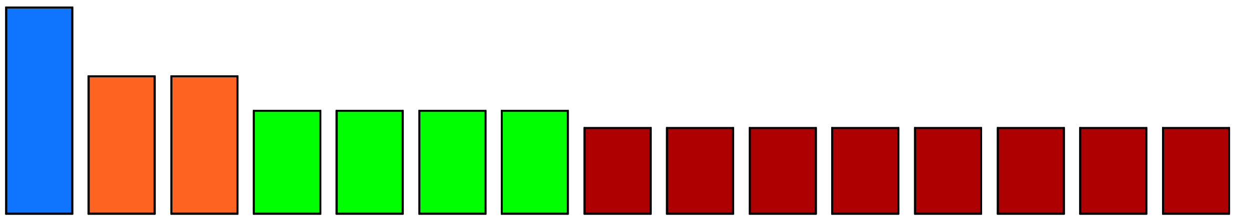
Result:



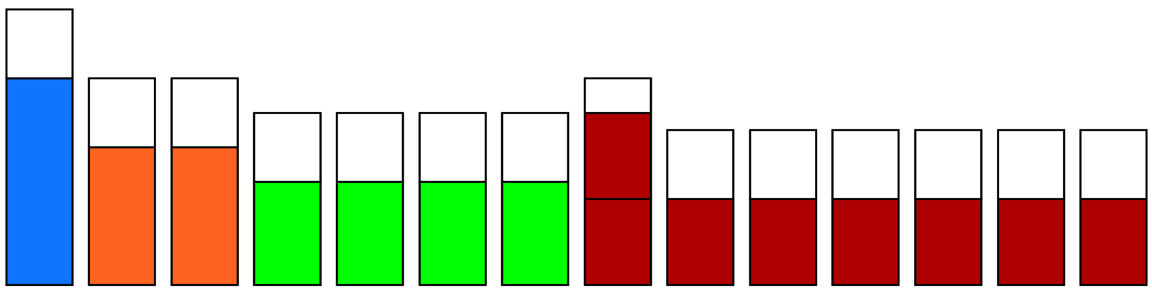
Asymptotically, FFI uses 2 times what OPT (FFD) uses.

FFD — First-Fit Decreasing

$B = 16$.
Input sizes: $[12], 2 \times [8], 4 \times [6], 8 \times [5]$
Bin sizes: $[16], 2 \times [12], 4 \times [10], [12], 6 \times [9]$



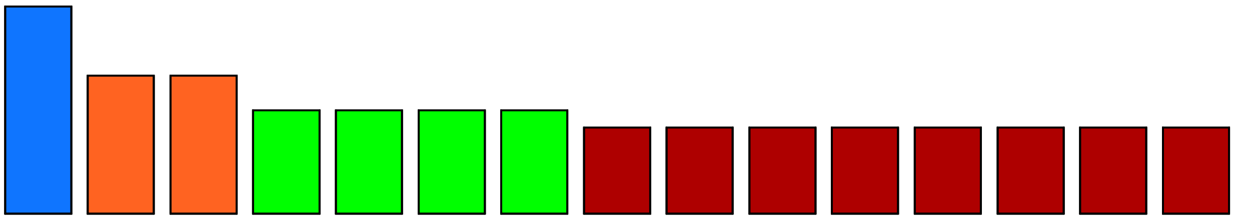
Result:



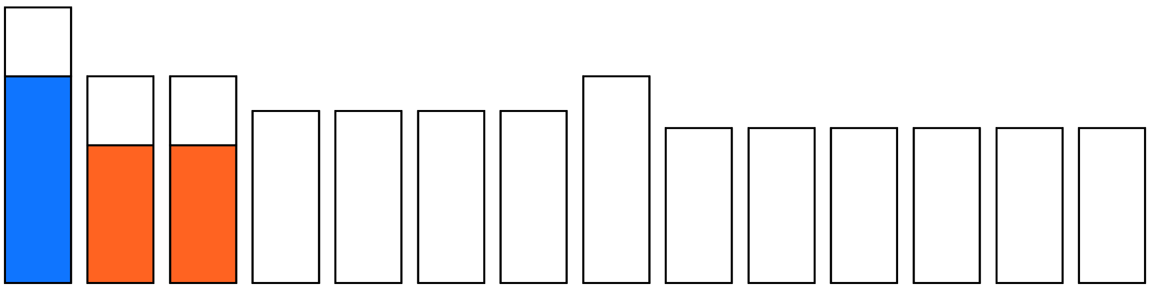
FFD uses ≈ 2 times what OPT uses. [G. Zhang]

$\mathrm{FFD}_{2/3}$ — First-Fit Decreasing$_{2/3}$

$B = 16$.
Input sizes: $[12], 2 \times [8], 4 \times [6], 8 \times [5]$
Bin sizes: $[16], 2 \times [12], 4 \times [10], [12], 6 \times [9]$



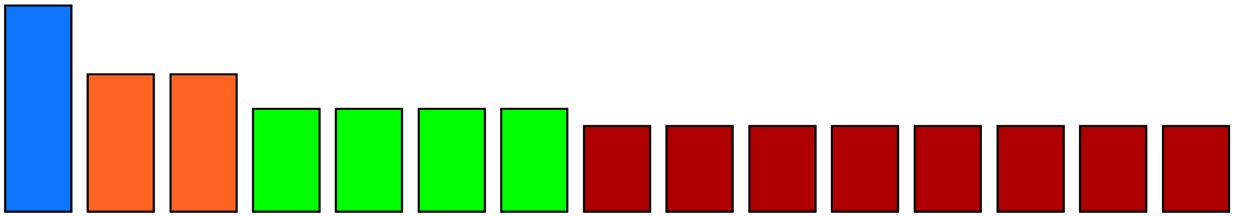
Partial result:



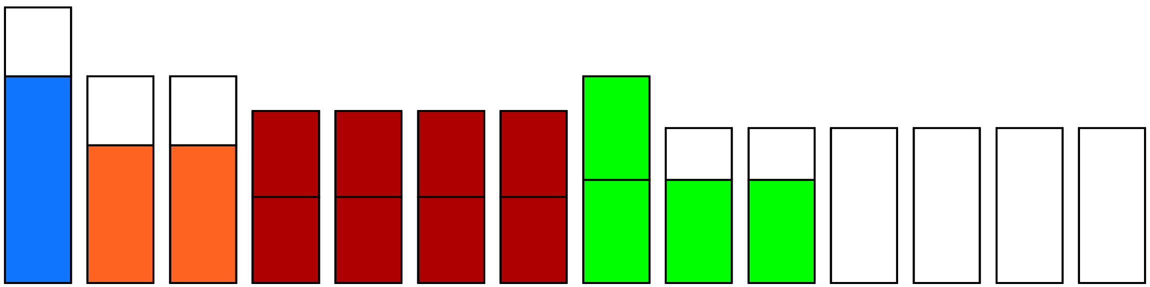
$\mathrm{FFD}_{2/3}$ treats items $[12], [8], [8]$ as FFD. But not $[6]$.

$FFD_{2/3}$ — First-Fit Decreasing$_{2/3}$

$B = 16$.
Input sizes: $[12], 2 \times [8], 4 \times [6], 8 \times [5]$
Bin sizes: $[16], 2 \times [12], 4 \times [10], [12], 6 \times [9]$



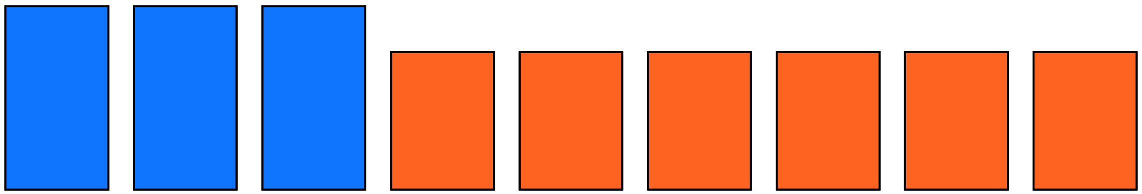
Result:



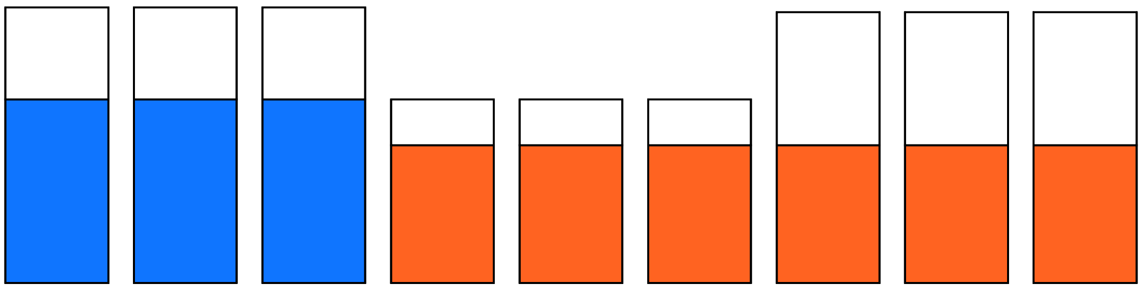
Items of size 5 paired in bins of size 10.

$\text{FFD}_{2/3}$ — First-Fit Decreasing$_{2/3}$

$B = 60$.
Input sizes: $n \times [40], 2n \times [30]$
Bin sizes: $n \times [60], n \times [40], n \times [59]$



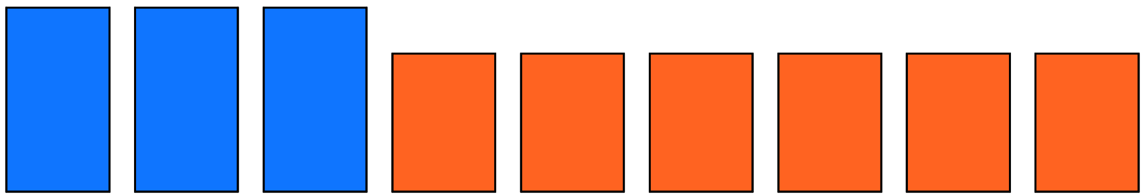
Result:



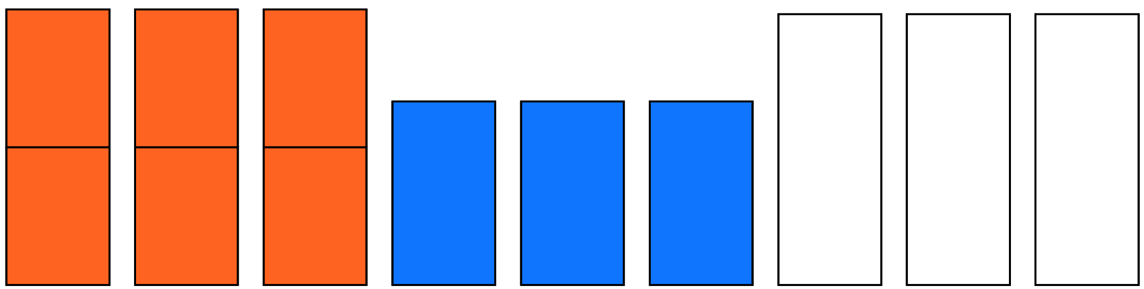
$\text{FFD}_{2/3}$ uses $n \times 159$.

$\mathbf{FFD}_{3/4}$ — First-Fit Decreasing$_{3/4}$

$B = 60$.
Input sizes: $n \times [40], 2n \times [30]$
Bin sizes: $n \times [60], n \times [40], n \times [59]$



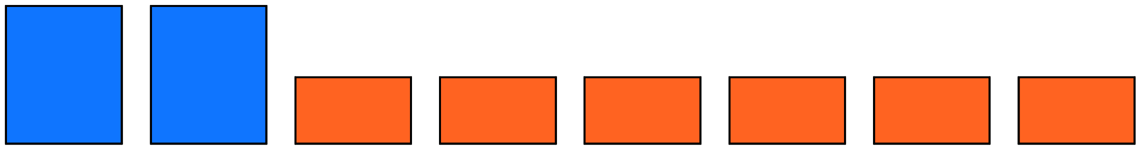
Result:



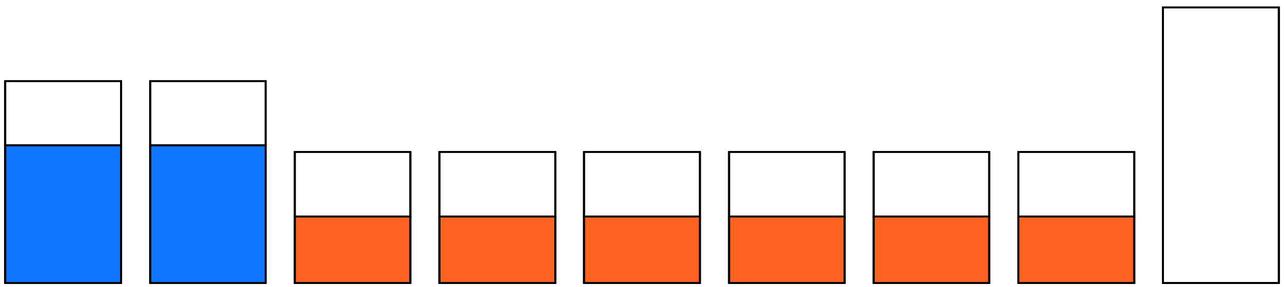
$\text{FFD}_{3/4}$ uses $n \times 100$. $\text{CR}_{FFD_\alpha} \geq \frac{2+\alpha}{1+\alpha}$.

$\text{FFD}_{2/3}$ — First-Fit Decreasing$_{2/3}$

$B = 120$.
Input sizes: $2n \times [60], 6n \times [29]$
Bin sizes: $2n \times [88], 6n \times [57], n \times [120]$



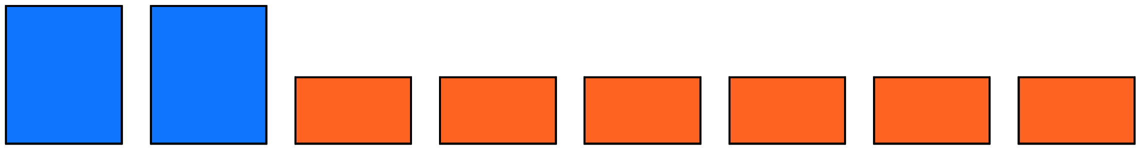
Result:



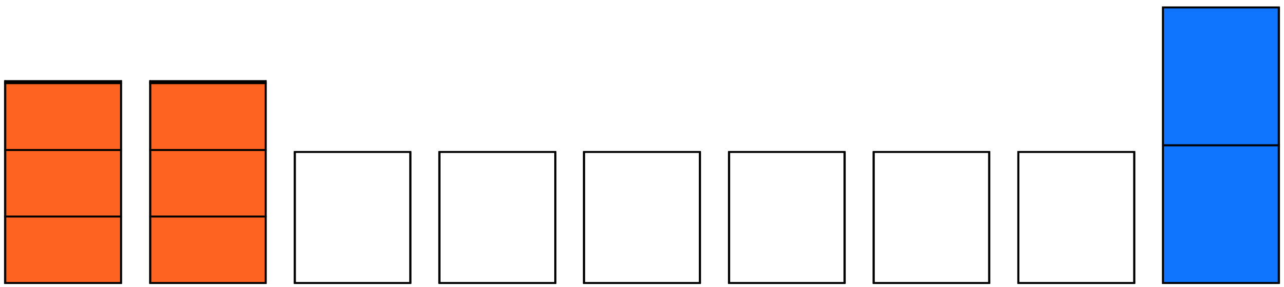
$\text{FFD}_{2/3}$ uses $n \times 518$.

$\mathbf{FFD}_{3/4}$ — First-Fit Decreasing$_{3/4}$

$B = 120$.
Input sizes: $2n \times [60], 6n \times [29]$
Bin sizes: $2n \times [88], 6n \times [57], n \times [120]$



Result:



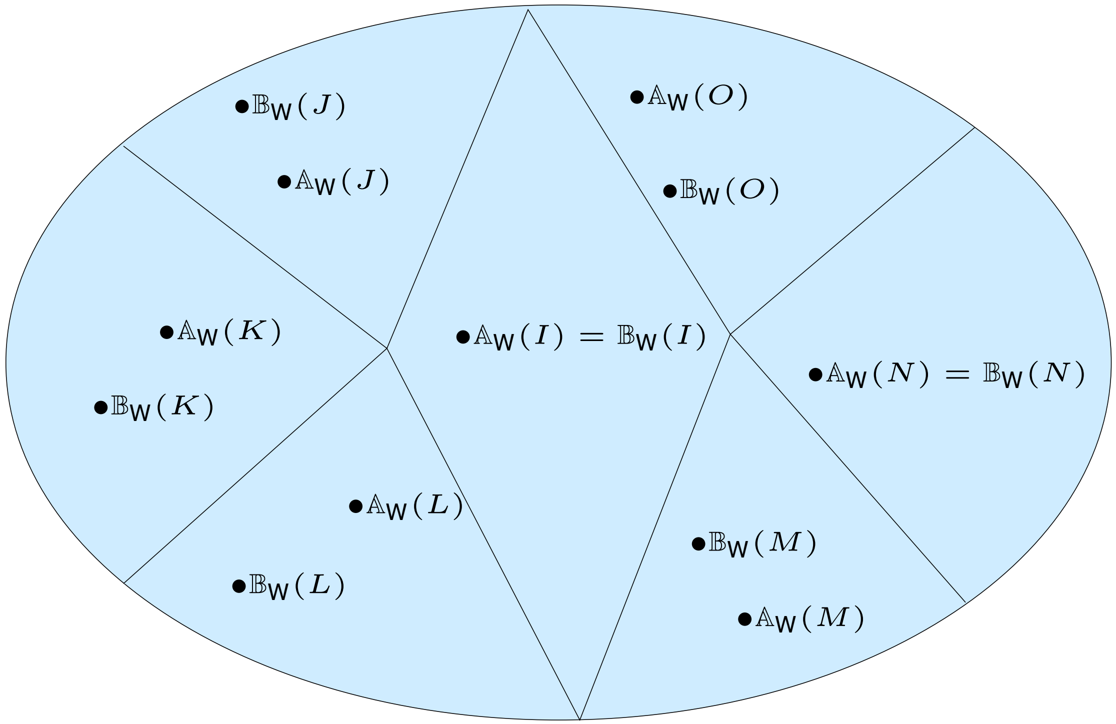
$\text{FFD}_{3/4}$ uses $n \times 416$. $\text{CR}_{FFD_{2/3}} \geq \frac{2(3s-2)+6(2s-3)}{2(3s-2)+4s} \approx 1.8$.

Competitive Ratio — Results

- $\mathsf{CR}_{\mathsf{FFI}} = \mathsf{CR}_{\mathsf{FFD}} = 2$. [G. Zhang]
- For $\alpha \le \frac{r-1}{r}$, $\frac{3r}{2r-1} \le \mathsf{CR}_{\mathsf{FFD}_\alpha}$.
- $1.8 \le \mathsf{CR}_{\mathsf{FFD}_{2/3}} \le 13/7 \approx 1.857$.
- $\mathsf{CR}_{\mathbb{A}} \le 2$ for any “reasonable” $\mathbb{A}$. [G. Zhang]
- $\mathsf{CR}_{\mathbb{A}} \ge 5/4$ for any deterministic $\mathbb{A}$.

Relative Worst Order Ratio

$\mathbb{A}_{\mathsf{W}}(I)$: $\mathbb{A}$'s performance on worst permutation of I, i.e.,
$\mathbb{A}_{\mathsf{W}}(I) = \max_{\sigma} \{\mathbb{A}(\sigma(I))\}$.



[Boyar,Favrholdt: CIAC 03]

If $\mathbb{A}_{\mathsf{W}}(I) \geq \mathbb{B}_{\mathsf{W}}(I) - b$ for all I,

$$\mathsf{WR}_{\mathbb{A},\mathbb{B}} = \inf \{c \mid \mathbb{A}_{\mathsf{W}}(I) \leq c \cdot \mathbb{B}_{\mathsf{W}}(I) + b \text{ for all } I\}.$$

Relative Worst Order Ratio

Competitive Ratio:

$$\mathsf{CR}_{\mathbb{A}} = \max_{I} \frac{\mathbb{A}(I)}{\mathsf{OPT}(I)}$$

Relative Worst Order Ratio:

$$\mathsf{WR}_{\mathbb{A}} = \max_{I} \frac{\max_{\sigma} \left\{ \mathbb{A}(\sigma(I)) \right\}}{\max_{\sigma} \left\{ \mathbb{B}(\sigma(I)) \right\}}$$

Relative Worst Order Ratio — Results

- FFD is better than FFI
- FFD_α is better than FFI
- FFD and FFD_α are incomparable

Open Problems

- Best α for FFD$_\alpha$?
- Exact competitive ratio of FFD$_\alpha$?
- Other algorithms?

Paging Results w. RWOR

- New algorithm RLRU - better than LRU
- LRU better than FWF
- Look-ahead helps

Other Results w. RWOR

Bin Packing:

Worst-Fit better than Next-Fit.

Dual Bin Packing:

First-Fit better than Worst-Fit.

Bin Coloring:

Greedy better than keeping only one open bin.

Scheduling – minimizing makespan on two related machines:

Post-Greedy better than using only fast machine.

Proportional Price Seat Reservation:

First-Fit better than Worst-Fit.