UNIVERSITÄT ZU LÜBECK

Fully Dynamic Bin Packing Revisited

Published in APPROX/RANDOM 2015

Sebastian Berndt [1] Klaus Jansen [2] Kim-Manuel Klein [2]

[1] Institut für Theoretische Informatik, Universität zu Lübeck

[2] Department of Computer Science, Christian-Albrechts-University to Kiel

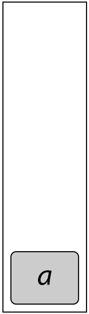
Fully Dynamic Bin Packing =
Online + Removal + Repacking

Fully Dynamic Bin Packing = Online + Removal + Repacking

INS: $a/0.2$

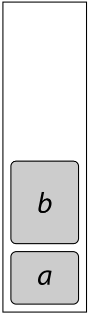
Fully Dynamic Bin Packing = Online + Removal + Repacking

INS: $a/0.2$, **INS**: $b/0.3$



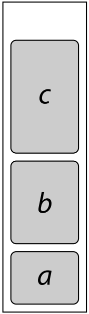
Fully Dynamic Bin Packing = Online + Removal + Repacking

INS: $a/0.2$, **INS**: $b/0.3$, **INS**: $c/0.4$



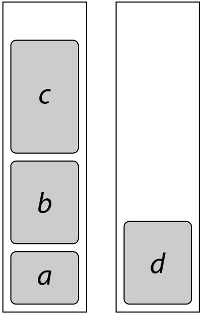
Fully Dynamic Bin Packing = Online + Removal + Repacking

INS: $a/0.2$, **INS**: $b/0.3$, **INS**: $c/0.4$, **INS**: $d/0.2$



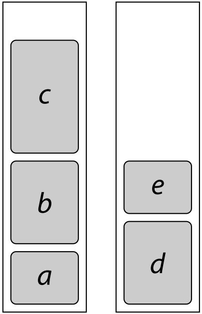
Fully Dynamic Bin Packing = Online + Removal + Repacking

INS: $a/0.2$, **INS**: $b/0.3$, **INS**: $c/0.4$, **INS**: $d/0.2$, **INS**: $e/0.3$



Fully Dynamic Bin Packing = Online + Removal + Repacking

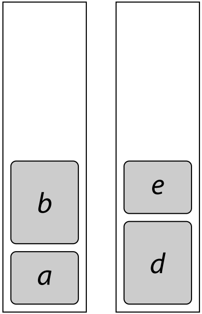
INS: $a/0.2$, **INS**: $b/0.3$, **INS**: $c/0.4$, **INS**: $d/0.2$, **INS**: $e/0.3$, **REM**: c



Sebastian Berndt , Klaus Jansen , Kim-Manuel Klein

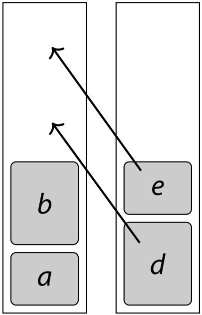
Fully Dynamic Bin Packing = Online + Removal + Repacking

INS: $a/0.2$, **INS**: $b/0.3$, **INS**: $c/0.4$, **INS**: $d/0.2$, **INS**: $e/0.3$, **REM**: c



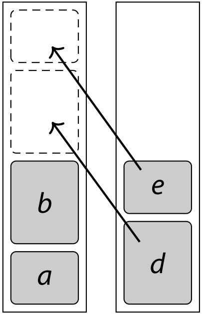
Fully Dynamic Bin Packing = Online + Removal + Repacking

INS: *a*/0.2, **INS**: *b*/0.3, **INS**: *c*/0.4, **INS**: *d*/0.2, **INS**: *e*/0.3, **REM**: *c*



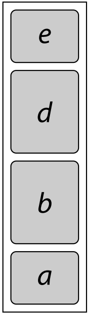
Fully Dynamic Bin Packing = Online + Removal + Repacking

INS: $a/0.2$, **INS**: $b/0.3$, **INS**: $c/0.4$, **INS**: $d/0.2$, **INS**: $e/0.3$, **REM**: c



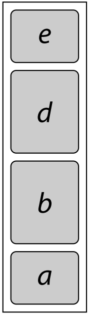
Fully Dynamic Bin Packing = Online + Removal + Repacking

INS: $a/0.2$, **INS**: $b/0.3$, **INS**: $c/0.4$, **INS**: $d/0.2$, **INS**: $e/0.3$, **REM**: c

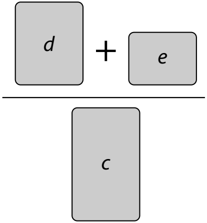


Fully Dynamic Bin Packing = Online + Removal + Repacking

INS: *a*/0.2, **INS**: *b*/0.3, **INS**: *c*/0.4, **INS**: *d*/0.2, **INS**: *e*/0.3, **REM**: *c*



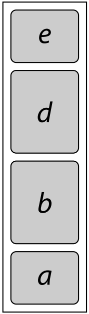
Migration Factor$_t$ =



$= \frac{\text{SIZE(moved)}}{\text{SIZE(new/removed)}}$

Fully Dynamic Bin Packing = Online + Removal + Repacking

INS: *a*/0.2, **INS**: *b*/0.3, **INS**: *c*/0.4, **INS**: *d*/0.2, **INS**: *e*/0.3, **REM**: *c*



Migration Factor$_t$ $= \frac{d + e}{c} = \frac{\text{SIZE(moved)}}{\text{SIZE(new/removed)}}$

Shifting Moves = number of moved items

Lower Bound

Migration Factor of $\Omega(1/\varepsilon)$ is necessary for ratio $1+\varepsilon$

$L = 1/2 - 1/9$(Migration Factor), $S = 1/3$(Migration Factor)

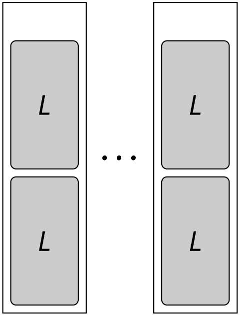
INS: L, **INS**: L, **INS**: $L, \ldots$

Lower Bound

Migration Factor of $\Omega(1/\varepsilon)$ is necessary for ratio $1+\varepsilon$

$L = 1/2 - 1/9(\text{Migration Factor})$, $\quad S = 1/3(\text{Migration Factor})$

INS: L, **INS**: L, **INS**: $L, \ldots$



Sebastian Berndt , Klaus Jansen , Kim-Manuel Klein

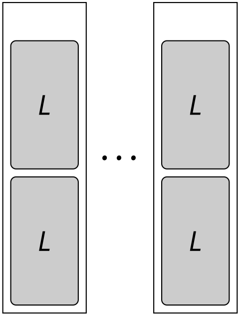
Lower Bound

Migration Factor of $\Omega(1/\varepsilon)$ is necessary for ratio $1+\varepsilon$

$L = 1/2 - 1/9(\text{Migration Factor})$, $S = 1/3(\text{Migration Factor})$

INS: L, **INS**: L, **INS**: $L, \ldots,$

INS: S, **INS**: S, **INS**: $S, \ldots$

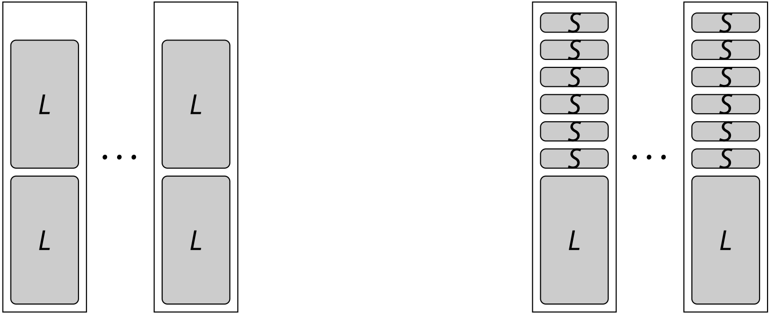


Migration Factor of $\Omega(1/\varepsilon)$ is necessary for ratio $1+\varepsilon$

$L = 1/2 - 1/9(\text{Migration Factor})$, $S = 1/3(\text{Migration Factor})$

INS: L, **INS**: L, **INS**: $L, \ldots,$

INS: S, **INS**: S, **INS**: $S, \ldots$



Known Results on upper bounds

Ratio	**REM**?	Shifting M.	Migration F.	Authors

Sebastian Berndt , Klaus Jansen , Kim-Manuel Klein

Known Results on upper bounds

Ratio	REM?	Shifting M.	Migration F.	Authors
3/2	✗	3	✗	Gambosi, Postiglione, Talamo (2000)
4/3	✗	7	✗	Gambosi, Postiglione, Talamo (2000)

Online Bin Packing with Repacking

Known Results on upper bounds

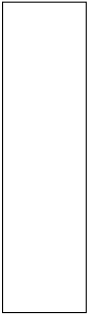
Ratio	**REM**?	Shifting M.	Migration F.	Authors
3/2	✗	3	✗	Gambosi, Postiglione, Talamo (2000)
4/3	✗	7	✗	Gambosi, Postiglione, Talamo (2000)
$1+\varepsilon$	✗	poly($\log n$) [am.]	✗	Ivković, Lloyd (1997)
$1+\varepsilon$	✗	$2^{\text{poly}(1/\varepsilon)}$	$2^{\text{poly}(1/\varepsilon)}$	Epstein, Levin (2006)
$1+\varepsilon$	✗	$\mathcal{O}(1/\varepsilon^4)$	$\mathcal{O}(1/\varepsilon^4)$	Jansen, Klein (2013)

Online Bin Packing with Repacking

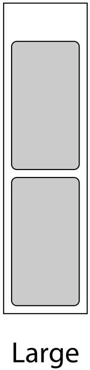
Known Results on upper bounds

Ratio	**REM**?	Shifting M.	Migration F.	Authors
3/2	✗	3	✗	Gambosi, Postiglione, Talamo (2000)
4/3	✗	7	✗	Gambosi, Postiglione, Talamo (2000)
$1+\varepsilon$	✗	poly($\log n$) [am.]	✗	Ivković, Lloyd (1997)
$1+\varepsilon$	✗	$2^{\text{poly}(1/\varepsilon)}$	$2^{\text{poly}(1/\varepsilon)}$	Epstein, Levin (2006)
$1+\varepsilon$	✗	$\mathcal{O}(1/\varepsilon^4)$	$\mathcal{O}(1/\varepsilon^4)$	Jansen, Klein (2013)
5/4	✓	poly($\log n$) [am.]	✗	Ivković, Lloyd (1998)
$1+\varepsilon$	✓	$\mathcal{O}(1/\varepsilon^4 \log(1/\varepsilon))$	$\mathcal{O}(1/\varepsilon^4 \log(1/\varepsilon))$	this work

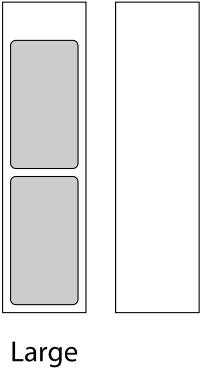
An Overview on the Packing



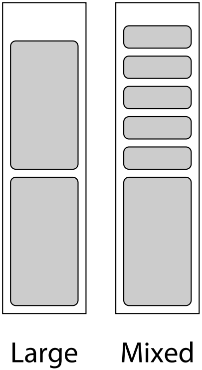
An Overview on the Packing



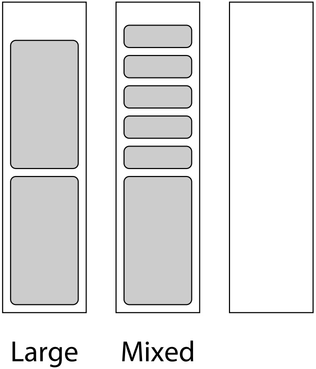
An Overview on the Packing



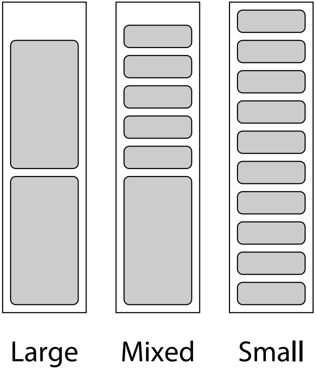
An Overview on the Packing



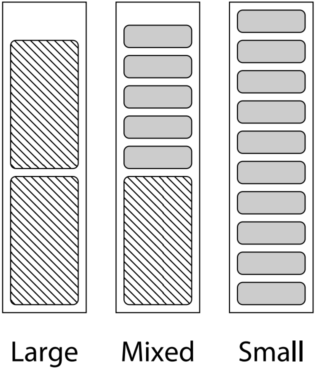
An Overview on the Packing



An Overview on the Packing

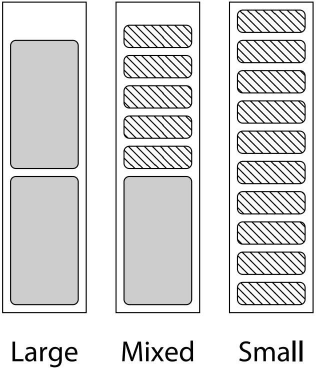


An Overview on the Packing



- Pack via LP

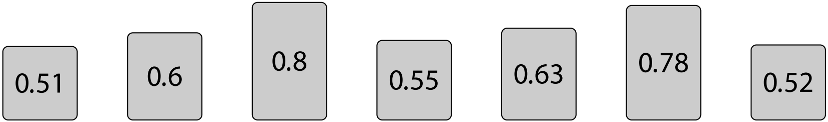
An Overview on the Packing



- Pack via LP
- Pack via “Sorting”

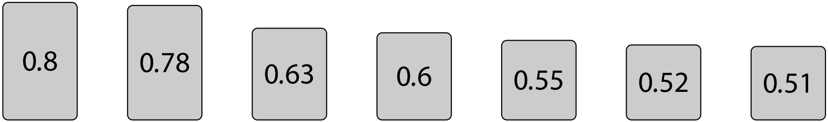
Rounding Large Items Geometrically

1. Find items of size (0.5, 1]



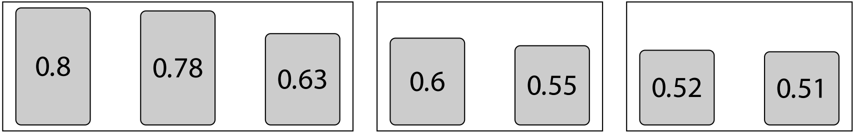
Rounding Large Items Geometrically

1. Find items of size (0.5, 1]
2. Sort items by size



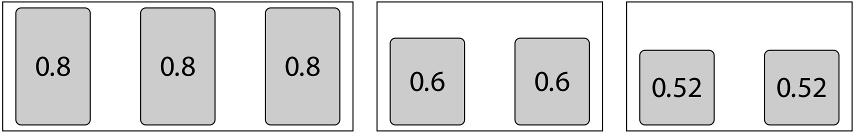
Rounding Large Items Geometrically

1. Find items of size (0.5, 1]
2. Sort items by size
3. Group items in lists



Rounding Large Items Geometrically

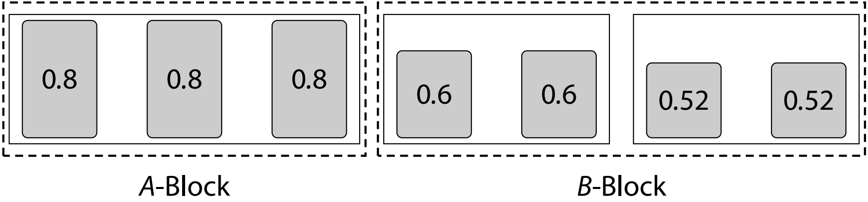
1. Find items of size (0.5, 1]
2. Sort items by size
3. Group items in lists
4. Round items



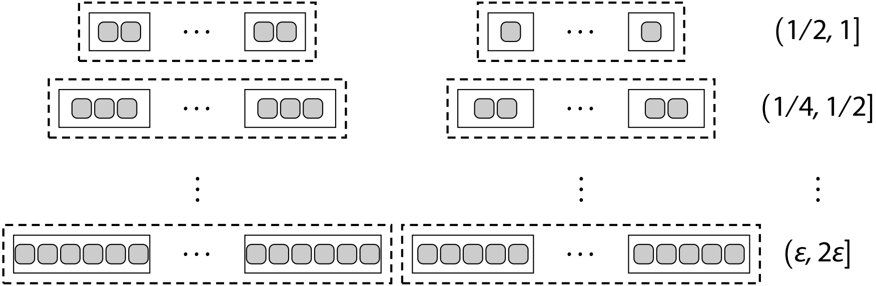
Sebastian Berndt , Klaus Jansen , Kim-Manuel Klein

Rounding Large Items Geometrically

1. Find items of size (0.5, 1]
2. Sort items by size
3. Group items in lists
4. Round items

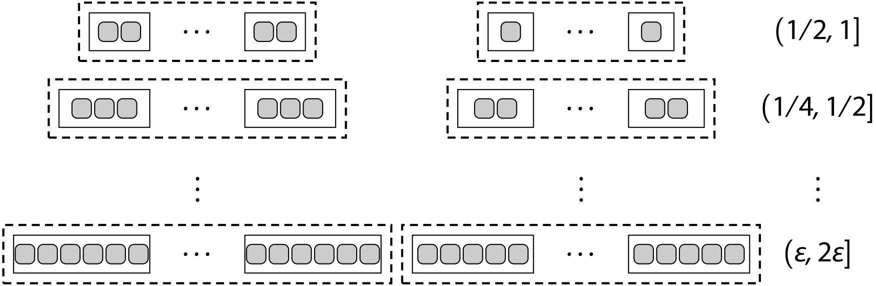


Packing Large Items



- Pack rounded items via LP

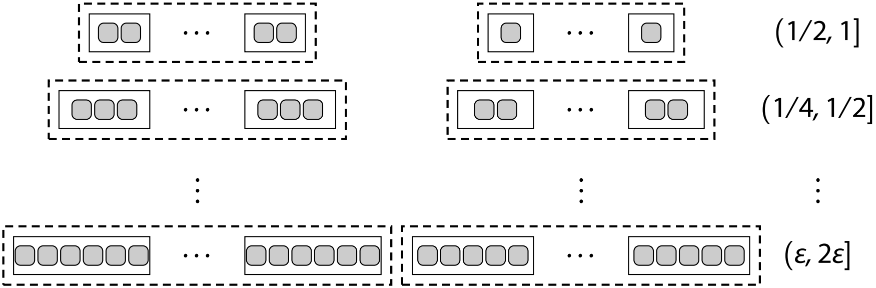
Packing Large Items



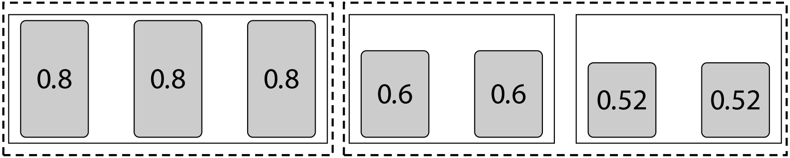
- Pack rounded items via LP
- Size of lists depends on volume of instance ⇒ Shifting

Sebastian Berndt , Klaus Jansen , Kim-Manuel Klein

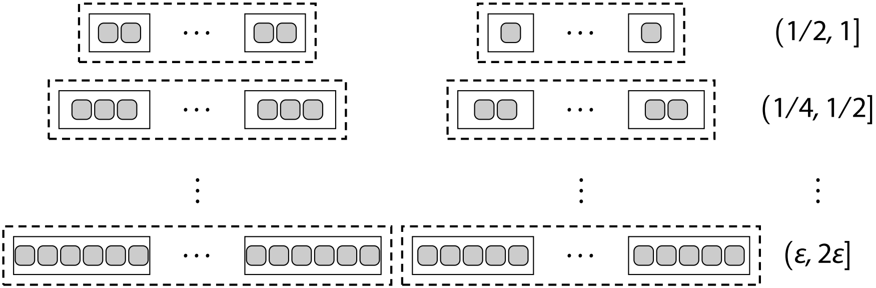
Packing Large Items



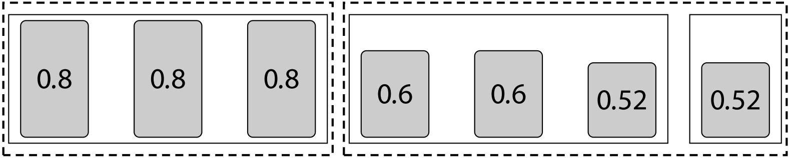
- Pack rounded items via LP
- Size of lists depends on volume of instance ⇒ Shifting



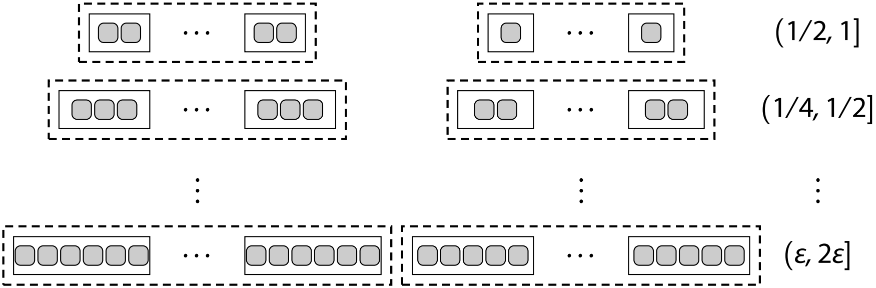
Packing Large Items



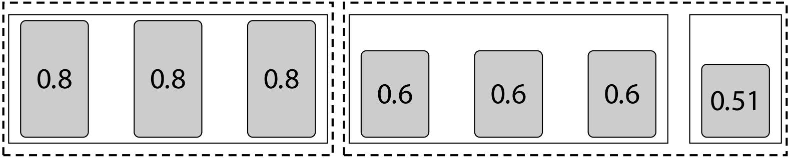
- Pack rounded items via LP
- Size of lists depends on volume of instance ⇒ Shifting



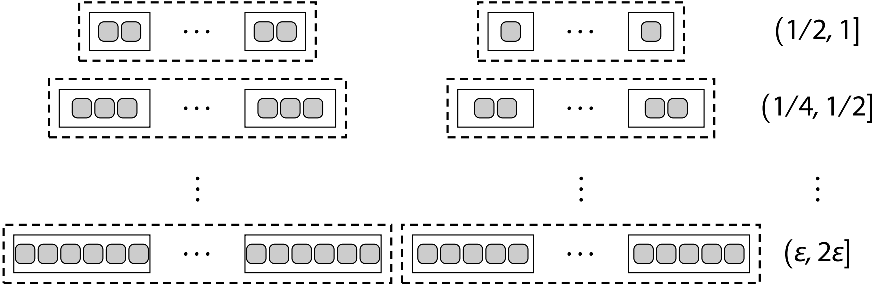
Packing Large Items



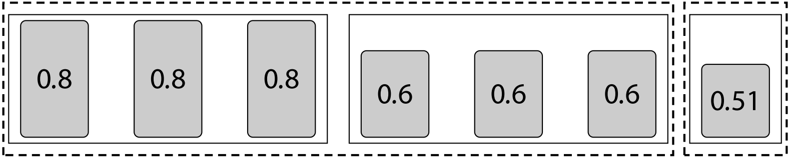
- Pack rounded items via LP
- Size of lists depends on volume of instance ⇒ Shifting



Packing Large Items



- Pack rounded items via LP
- Size of lists depends on volume of instance ⇒ Shifting



Sebastian Berndt , Klaus Jansen , Kim-Manuel Klein

Packing Small Items is Difficult

Greedy fails: $\varepsilon \gg L \gg S$

Packing Small Items is Difficult

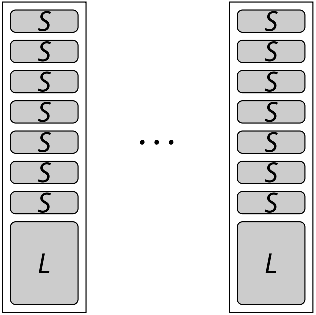
Greedy fails: $\varepsilon \gg L \gg S$

$(\texttt{INS}: L, \texttt{INS}: S, \ldots, \texttt{INS}: S)^*$

Packing Small Items is Difficult

Greedy fails: $\varepsilon \gg L \gg S$

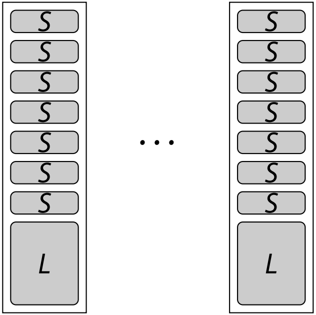
$\left(\texttt{INS}: L, \texttt{INS}: S, \ldots, \texttt{INS}: S\right)^*$



Packing Small Items is Difficult

Greedy fails: $\varepsilon \gg L \gg S$

$(\mathbf{INS}: L, \mathbf{INS}: S, \ldots, \mathbf{INS}: S)^*$

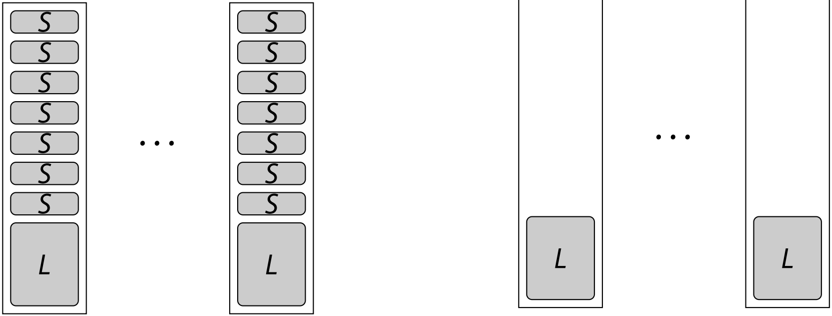


REM: S, **REM**: S, ...

Packing Small Items is Difficult

Greedy fails: $\varepsilon \gg L \gg S$

$(\texttt{INS}: L, \texttt{INS}: S, \ldots, \texttt{INS}: S)^*$ $\texttt{REM}: S, \texttt{REM}: S, \ldots$



Packing Small Items via Sorting

Idea: "Sort" small items from left to right

Packing Small Items via Sorting

Idea: “Sort” small items from left to right

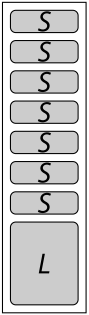
$$\texttt{INS}: L, \texttt{INS}: S, \ldots, \texttt{INS}: S,$$

Sebastian Berndt , Klaus Jansen , Kim-Manuel Klein

Packing Small Items via Sorting

Idea: "Sort" small items from left to right

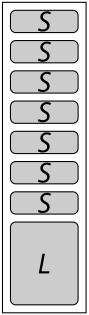
INS: L, **INS**: S,. . . , **INS**: S,



Packing Small Items via Sorting

Idea: “Sort” small items from left to right

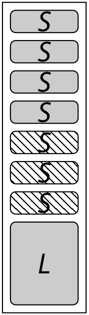
INS: L, **INS**: S,. . . , **INS**: S, **INS**: L



Packing Small Items via Sorting

Idea: "Sort" small items from left to right

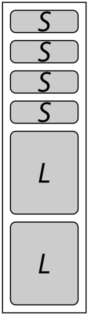
INS: L, **INS**: S,. . . , **INS**: S, **INS**: L



Packing Small Items via Sorting

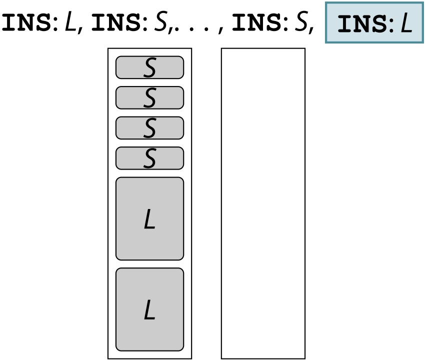
Idea: "Sort" small items from left to right

INS: L, **INS**: S,. . . , **INS**: S, **INS**: L



Packing Small Items via Sorting

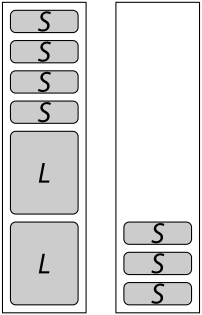
Idea: “Sort” small items from left to right



Packing Small Items via Sorting

Idea: "Sort" small items from left to right

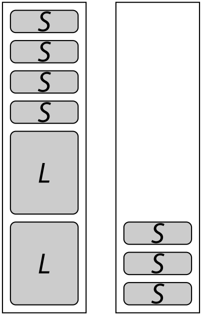
INS: L, **INS**: S,. . . , **INS**: S, **INS**: L



Packing Small Items via Sorting

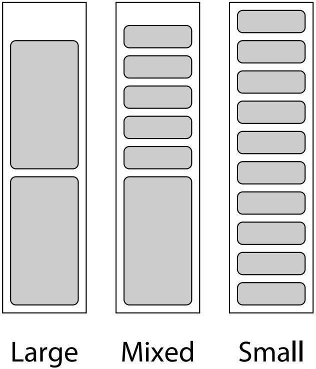
Idea: "Sort" small items from left to right

INS: L, **INS**: S,. . . , **INS**: S, **INS**: L



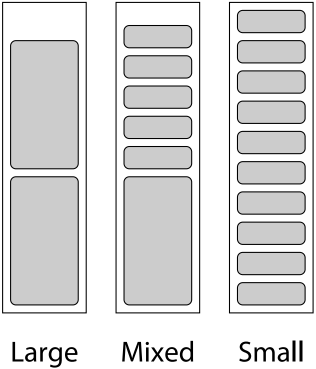
Stop at every $1/\varepsilon$-th bin (buffer bin) to bound Migration

An Overview on the Packing



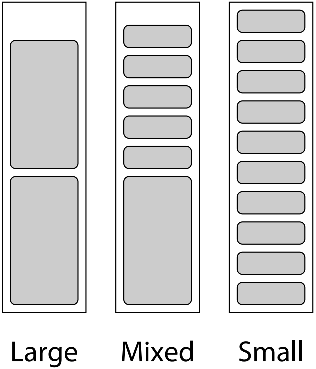
- Pack large items via LP

An Overview on the Packing



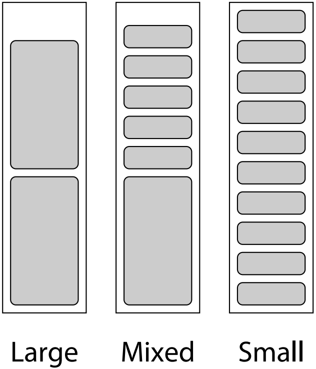
- Pack large items via LP
- Pack small items via "Sorting"

An Overview on the Packing



- Pack large items via LP
- Pack small items via "Sorting"
- Small bins $\Rightarrow$ little free space in other bins

An Overview on the Packing

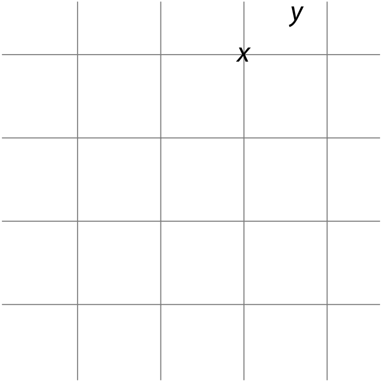


- Pack large items via LP
- Pack small items via “Sorting”
- Small bins $\Rightarrow$ little free space in other bins
- Relate nearly full/empty bins via potential function

Some Details

Theorem (Jansen and Klein '13)

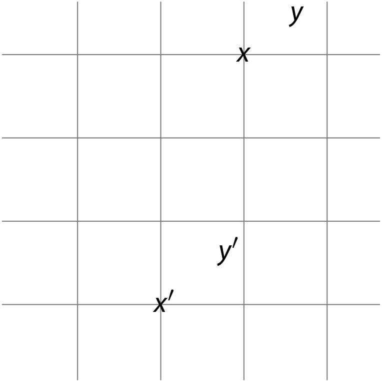
[Some requirements] There is an algorithm that returns for a LP/ILP pair (y, x) an α-improved LP/ILP pair (x', y') with $||x'|| \leq ||x|| - \alpha$ and $||y'|| \leq ||y|| - \alpha$.



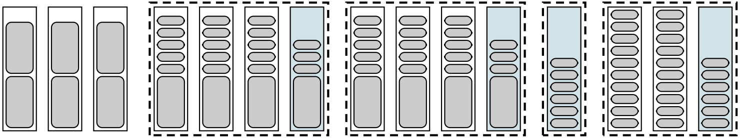
Some Details

Theorem (Jansen and Klein '13)

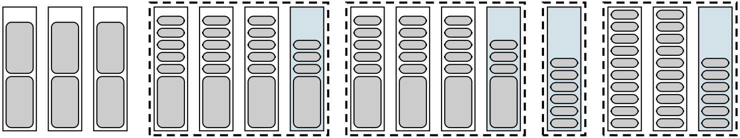
[Some requirements] There is an algorithm that returns for a LP/ILP pair (y, x) an α-improved LP/ILP pair (x', y') with $||x'|| \leq ||x|| - \alpha$ and $||y'|| \leq ||y|| - \alpha$.



Some Details

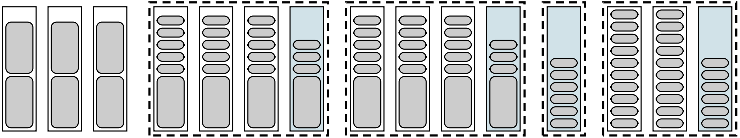


Some Details



Invariants:

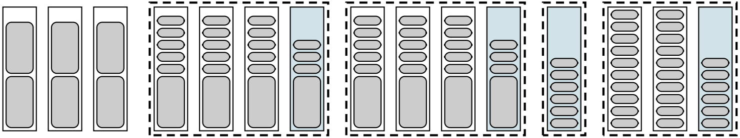
Some Details



Invariants:

- Small Items are sorted from left to right

Some Details

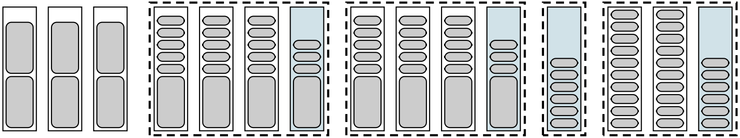


Invariants:

- Small Items are sorted from left to right
- Only buffer bins and large bins are non-full

Sebastian Berndt , Klaus Jansen , Kim-Manuel Klein

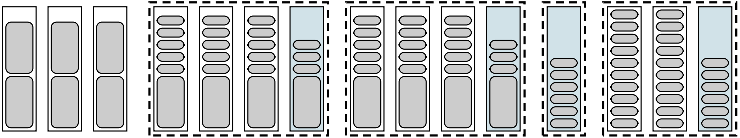
Some Details



Invariants:

- Small Items are sorted from left to right
- Only buffer bins and large bins are non-full
- Length of each queue is in $[1/\varepsilon, 2/\varepsilon]$

Some Details



Invariants:

- Small Items are sorted from left to right
- Only buffer bins and large bins are non-full
- Length of each *queue* is in $[1/\varepsilon, 2/\varepsilon]$
- Number of non-filled large bins is $\approx$

$$\Phi = \sum_i r_i + \varepsilon \cdot \Delta + \ell$$

bins (↓ Δ)

fill-ratio of bb_i (↑ r_i)

mixed bins (↑ ℓ)

Questions Left

- Can we improve the lower bound on MF from $\Omega(1/\varepsilon)$?
  (**REM**-operation, $\omega(1)$ sizes)

Questions Left

- Can we improve the lower bound on MF from $\Omega(1/\varepsilon)$? (**REM**-operation, $\omega(1)$ sizes)
- Can we use ideas from offline bin packing to improve the running time from $\mathcal{O}(1/\varepsilon^6 \cdot \log^4(1/\varepsilon) \cdot n)$? (Plotkin and Shmoys and Tardos; Shachnai and Yehezkely; Jansen and Kraft)

Questions Left

- Can we improve the lower bound on MF from $\Omega(1/\varepsilon)$? (**`REM`**-operation, $\omega(1)$ sizes)
- Can we use ideas from offline bin packing to improve the running time from $\mathcal{O}(1/\varepsilon^6 \cdot \log^4(1/\varepsilon) \cdot n)$? (Plotkin and Shmoys and Tardos; Shachnai and Yehezkely; Jansen and Kraft)
- Can we simplify the handling of the small items?

Questions Left

- Can we improve the lower bound on MF from $\Omega(1/\varepsilon)$? (**REM**-operation, $\omega(1)$ sizes)
- Can we use ideas from offline bin packing to improve the running time from $\mathcal{O}(1/\varepsilon^6 \cdot \log^4(1/\varepsilon) \cdot n)$? (Plotkin and Shmoys and Tardos; Shachnai and Yehezkely; Jansen and Kraft)
- Can we simplify the handling of the small items?
- Can we adapt our techniques to other problems?