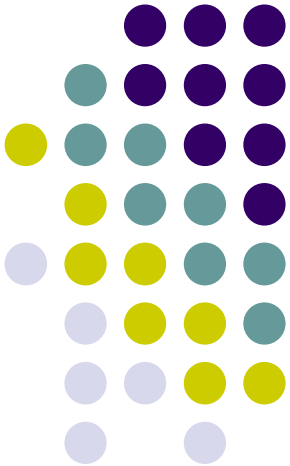
אוניברסיטת חיפה
University of Haifa
جامعة حيفا

Online bin packing of squares and cubes



Loay Mualem
Department of Computer Science
University of Haifa

Joint work with:
Leah Epstein
University of Haifa

Online bin packing problem

Input:

- A sequence of n items, each item i of size $s_i \in (0,1]$

Output:

- Pack the items into minimum number of bins such that each bin can contain items of sum at most 1.

Online bin packing algorithms

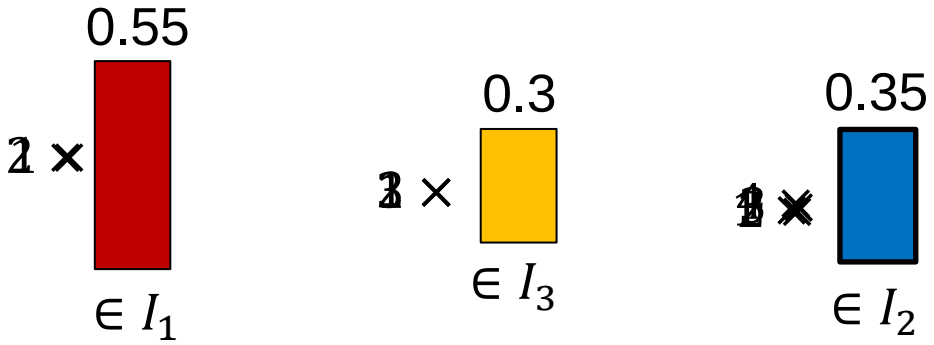
Analysis:

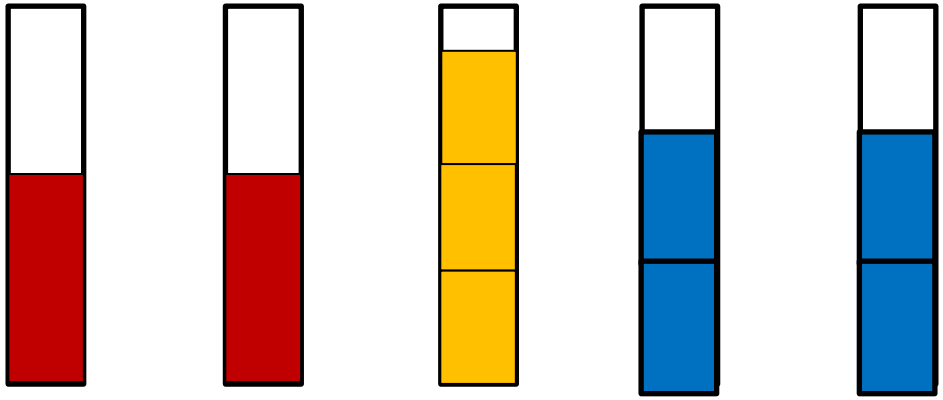
- Online bin packing algorithms are analyzed via $asymptotic\ competitive\ ratio$.
- Given an input list L, let $ALG(L)$ be the number of bins used (cost) obtained by applying algorithm ALG on the input L and let OPT be an optimal offline algorithm that uses minimum number of bins for packing the items. Let $OPT(L)$ denote the number of bins.
- An algorithm is $asymptotically\ r\text{-}competitive$ if there exists a constant C such that for any input L, $ALG(L) \leq r \cdot OPT(L) + C$.

Harmonic algorithm

- Partition the unit interval $(0,1]$ to M disjoint sub-intervals of the $I_j = \left(\frac{1}{j+1}, \frac{1}{j}\right]$ form $i = 1, \ldots, M-1$, and the Mth sub-interval is $(0, \frac{1}{M}]$.
- Given a list $S = (\ell_1, \ell_2, \ldots, \ell_n)$, where $\ell_i \in (0,1]$ for all i, an item ℓ_i is called an item of type j if $\ell_i \in I_j$.
- A bin is called $type\ i$ if it packs items of type i only.
- Each bin of type i for $1 \leq i \leq M$ can accommodate at most i items of type i.
- A bin is called $active$ if it still can receive items. Otherwise, it is considered to be $closed$.
- At each step of the algorithm, there is at most one active bin for each type.

- Input A:

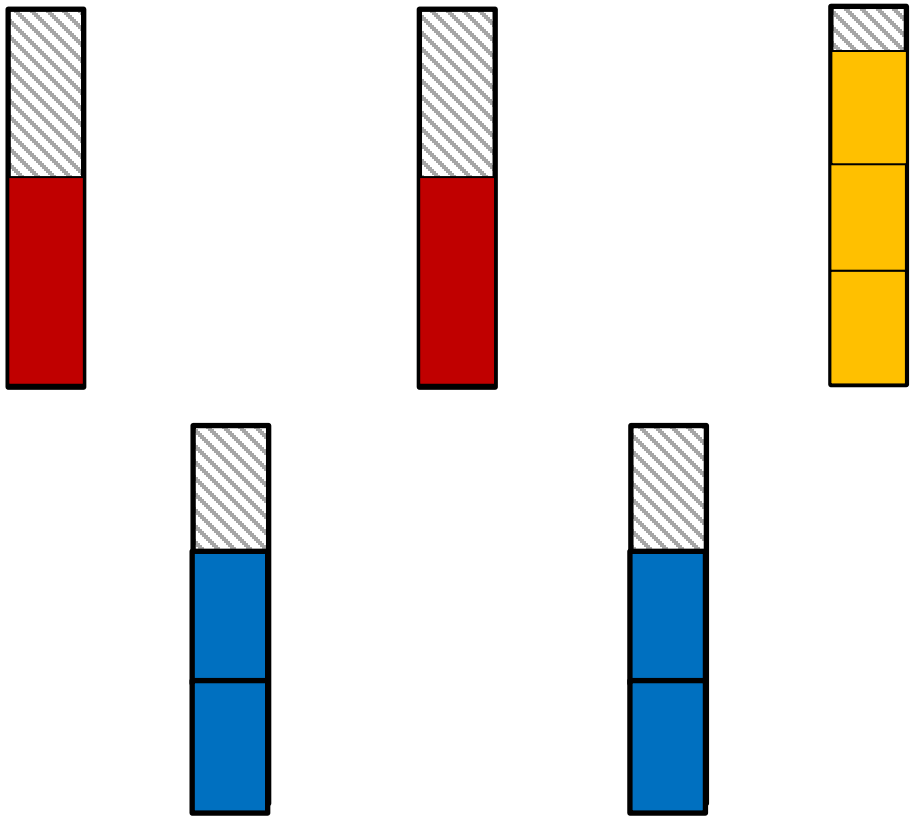




A lot of wasted space!

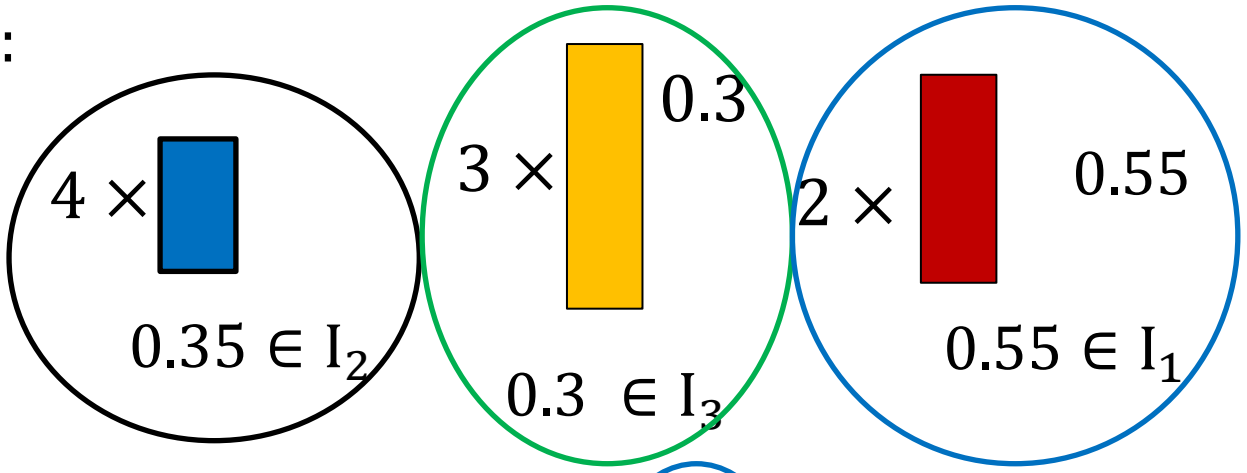
wasted space

- Harmonic output:



Analyzing Harmonic Algorithm

- The asymptotic competitive ratio for the Harmonic algorithm is analyzed via *weighting function*.
- The weight of every item of *type* i for $i = 1, \dots, M-1$ is simply $\frac{1}{i}$, for $i = \mathrm{M}$ there is different weight function.
- Weight of Input A:



$$Harmonic(A) = 4 \times \frac{1}{2} + 3 \times \frac{1}{3} + 2 \times \frac{1}{1} = \frac{4}{2} + \frac{3}{3} + \frac{2}{1} = 5$$

Online d-dimensional packing

Input:

A list L of items, where each item is a d-dimensional box, and in each dimension, the side length of an item does not exceed 1.

Output:

A packing of all input items of L into d-dimensional hypercube bins.

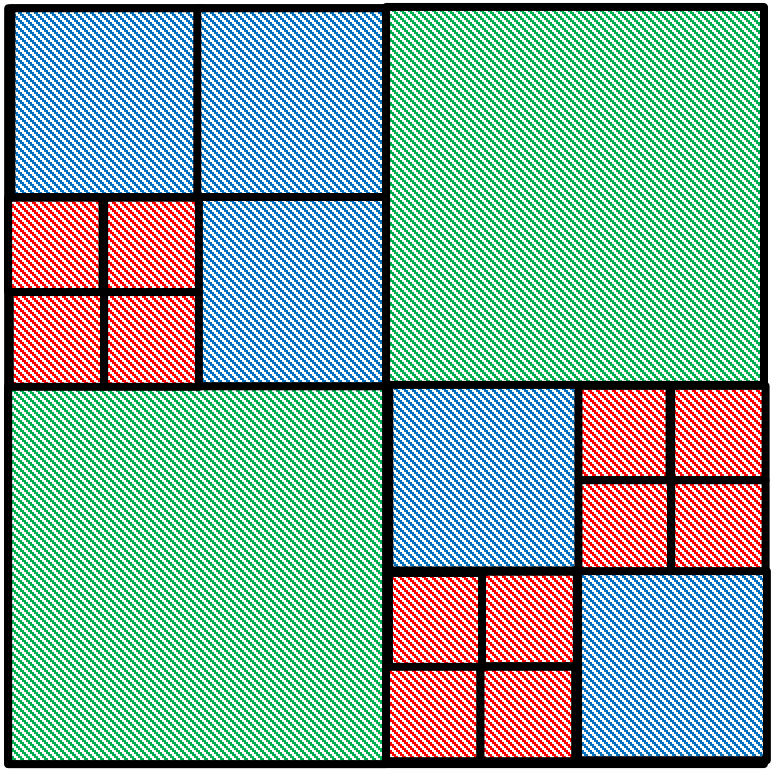
- No two items in a bin overlap with each other.
- The sides of item are parallel to sides of bins.

Example

$\frac{1}{8}$ 

$\frac{1}{4}$ 

$\frac{1}{2}$ 



Previous results

- 2.6875 and 6.25 for square and cube packing by Coppersmith and Raghavan (1989)
- 2.438272 for square packing by Seiden and van Stee (2003)
- 3.954 for cube packing by Miyazawa and Wakabayashi (2003)
- 2.24437 and 2.9421 for square and cube packing respectively by Epstein and van Stee (2005)
- 2.1187 and 2.6161 for square and cube packing respectively by Han et al. (2010)

Not valid.

2 counter examples	1	2
for square packing:	2.122946	2.120087899

Our results

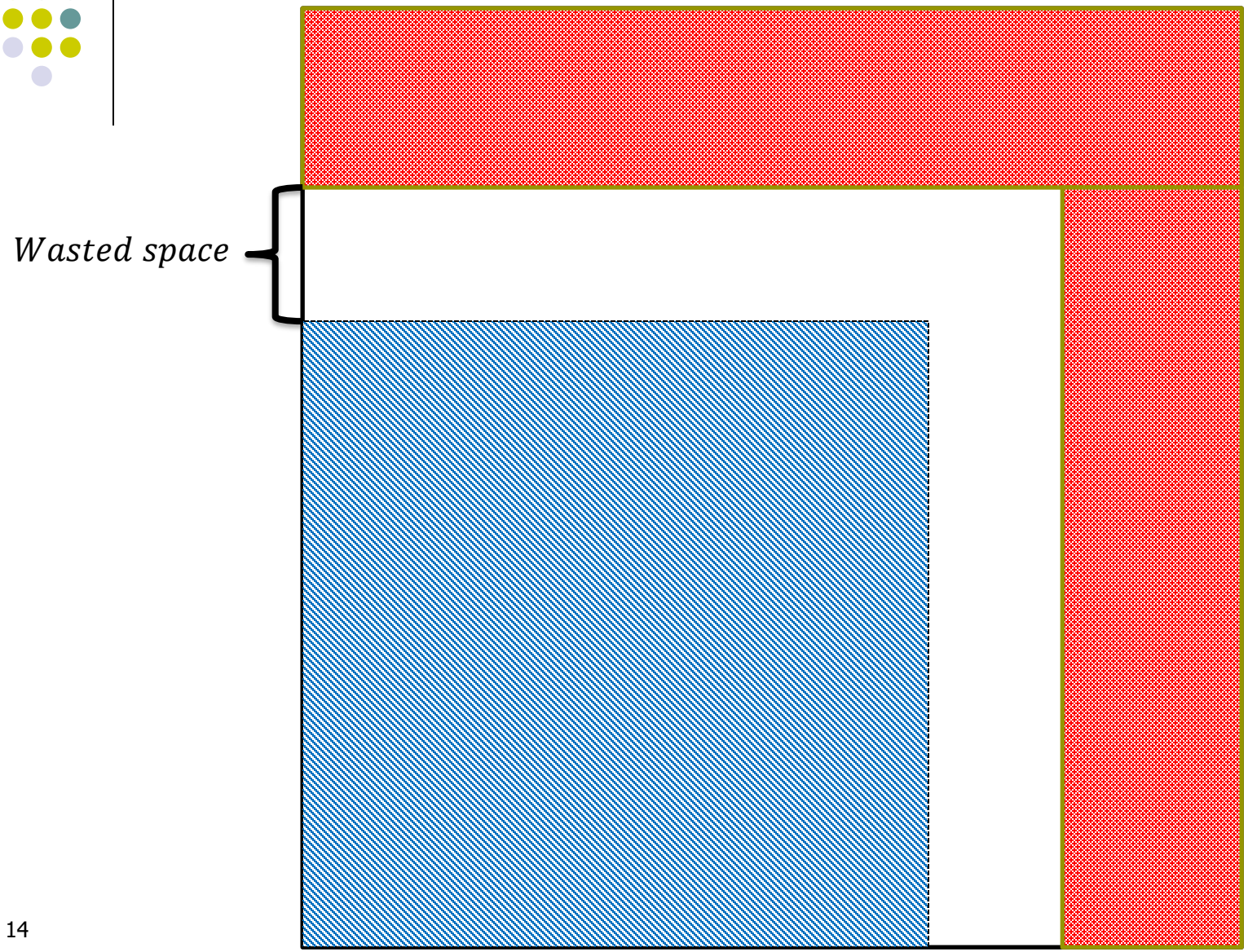
1. A new upper bound for asymptotic competitive ratio for square packing of 2.0885.
2. A new upper bound for asymptotic competitive ratio for cube packing of 2.573.

Extended Harmonic algorithm

- Unlike harmonic algorithm, $I_j = (\frac{1}{t_{j+1}}, \frac{1}{\mathrm{t_j}}]$ for $j \in \{1, \dots, n\}$.
- Each item ℓ_i is called type i item if $size(\ell_i) \in I_i$.
- Each item ℓ_i is considered small if $size(\ell_i) \leq \frac{1}{t_{n+1}}$ otherwise large.
- Pack small items by a different algorithm, pack large items via $Extended\ Harmonic$ algorithm.

Extended Harmonic

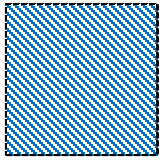
- Each incoming item gets a color, red or blue.
- Each type i of items has a fraction of α_i items colored red, and a fraction of $1 - \alpha_i$ items colored blue.
- Each bin can contain:
  1. Only blue items of type j
  2. Blue items of type j and red items of type i
  3. Only red items of type i


Wasted space

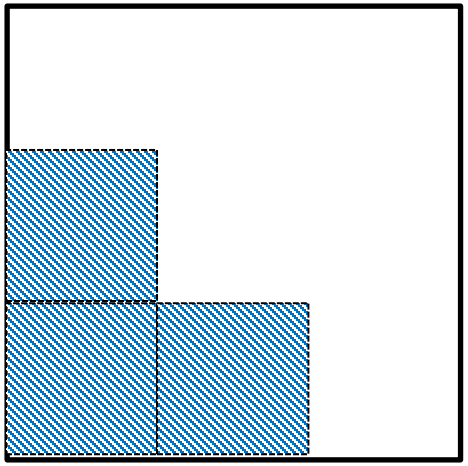
Packing blue items

Each blue item b of type i will be packed in the first open bin from the following ordered list of bins:

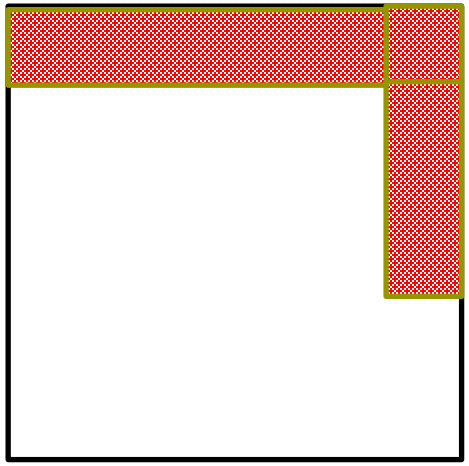
1. A bin which includes blue items of type i and there is enough space to pack the item b.
2. A bin which includes only red items of type i such that items type j are accepted as blue items in bins which includes blue items of type i
3. Open a new bin.



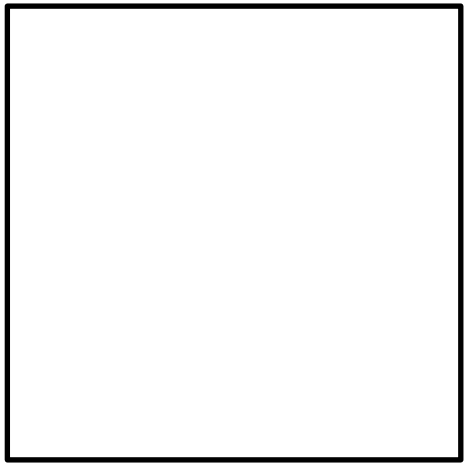
Priority 1



Priority 2



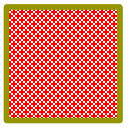
Priority 3



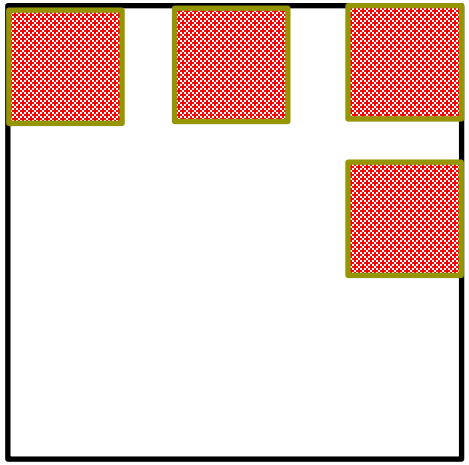
Packing red items

Each red item r of type i will be packed in the first open bin from the following ordered list of bins:

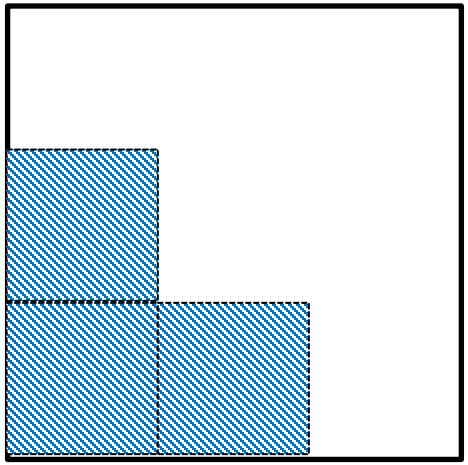
1. A bin which includes red items of type i and there is enough space to pack the item r
2. A bin which includes only blue items of type j such that items type i are accepted as red items in bins which includes blue items of type j
3. Open a new bin.



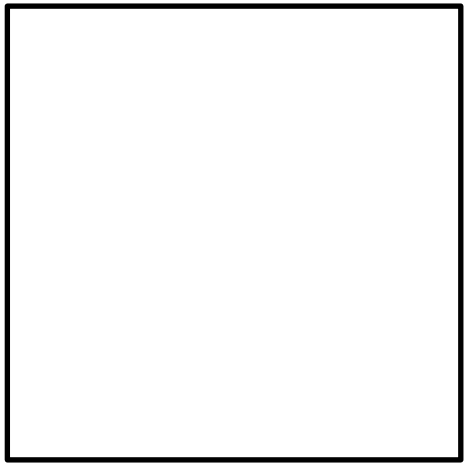
Priority 1



Priority 2



Priority 3



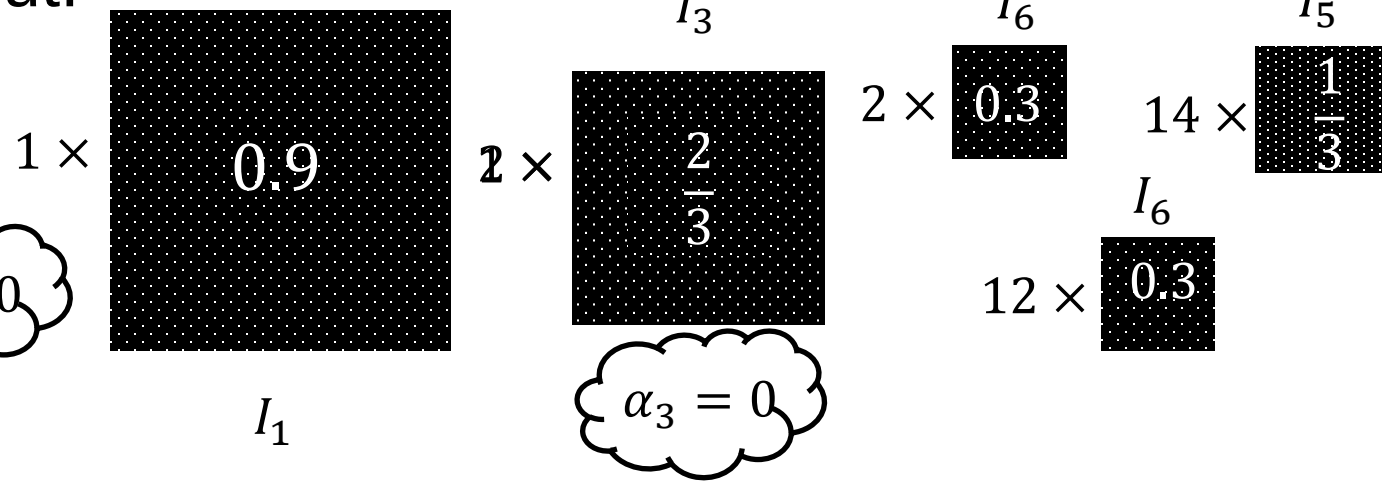
Extended Harmonic example

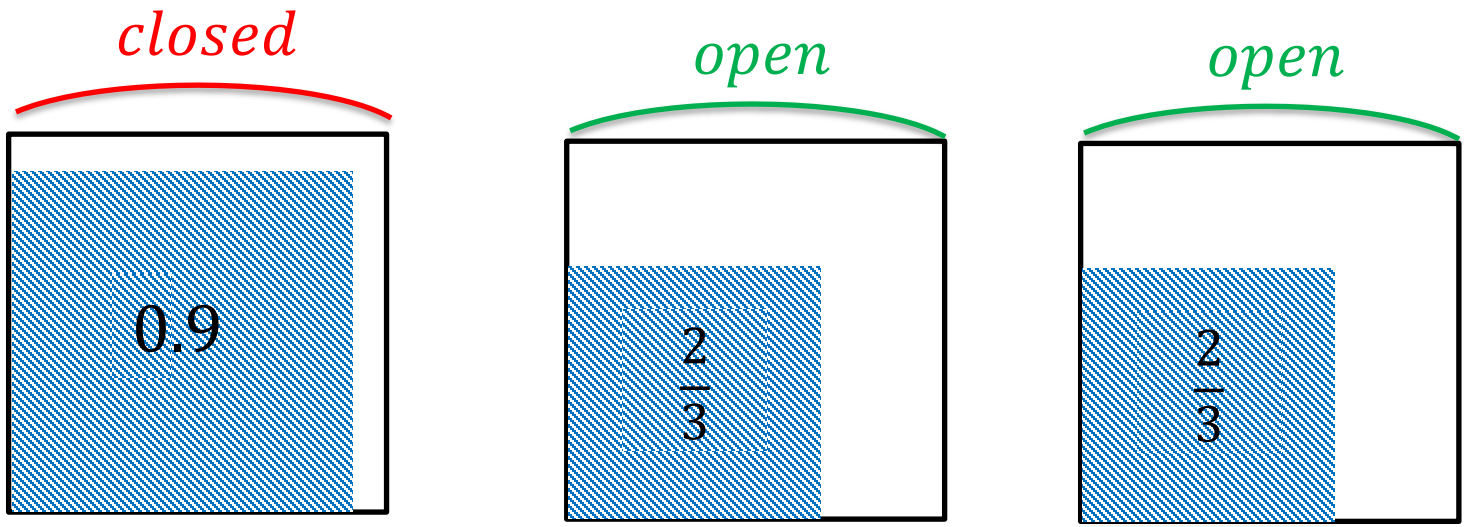
i	$(t_{i+1}, t_i]$	**Intervals accepted as red items**	α_i
1	$(0.7,1]$	-	0
2	$(\frac{2}{3}, 0.7]$	6	0
3	$(\frac{1}{2}, \frac{2}{3}]$	5,6	0
4	$(\frac{1}{3}, \frac{1}{2}]$	-	0
5	$(0.3, \frac{1}{3}]$	-	0.4
6	$(0.1,0.3]$	-	0.4

REMINDER:
α_i = fraction of red items type i

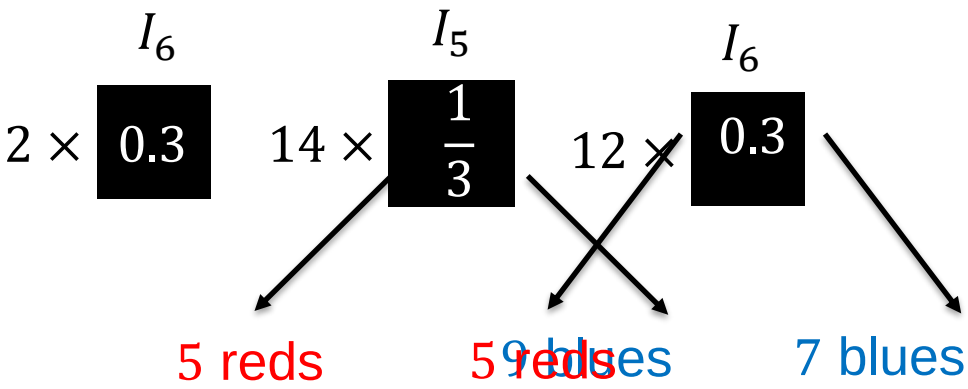
Note that in this example every item ℓ_i that satisfies $size(\ell_i) \leq 0.1$ is considered small item.

Input:


I_3
I_6
I_5
$1 \times$
0.9
$2 \times$
$\frac{2}{3}$
$2 \times$
0.3
$14 \times$
$\frac{1}{3}$
I_6
$12 \times$
0.3
$\alpha_1 = 0$
I_1
$\alpha_3 = 0$

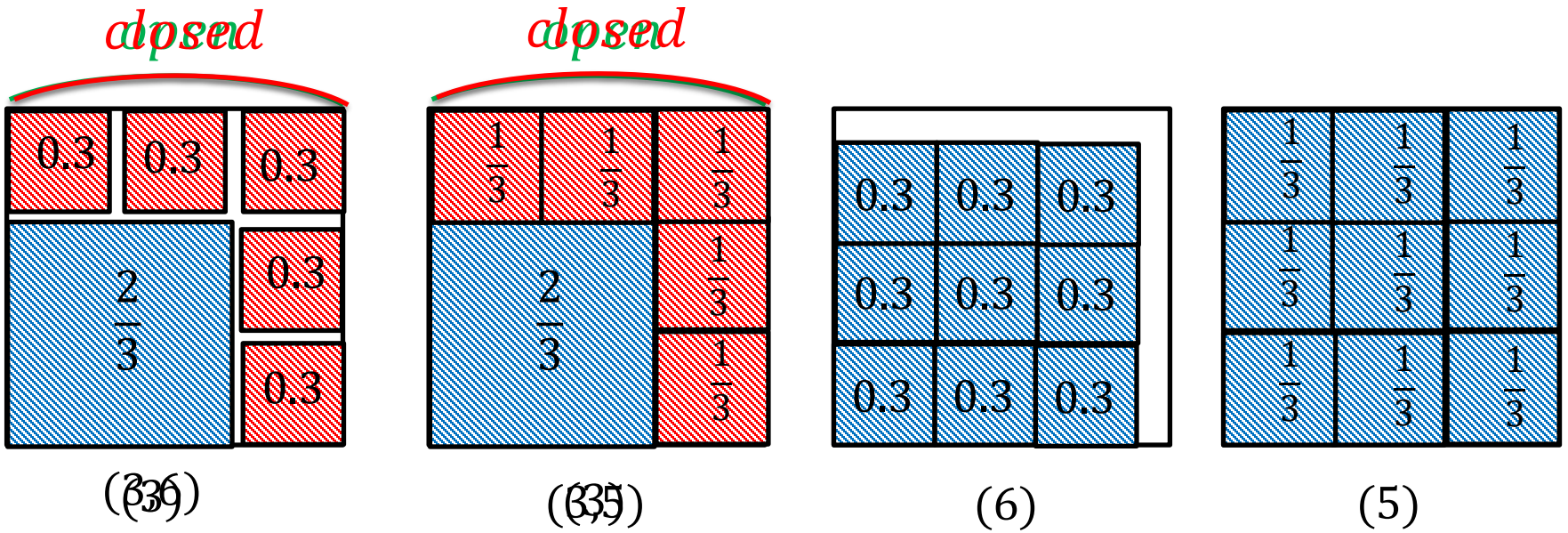

closed
open
open
0.9
$\frac{2}{3}$
$\frac{2}{3}$

Input:

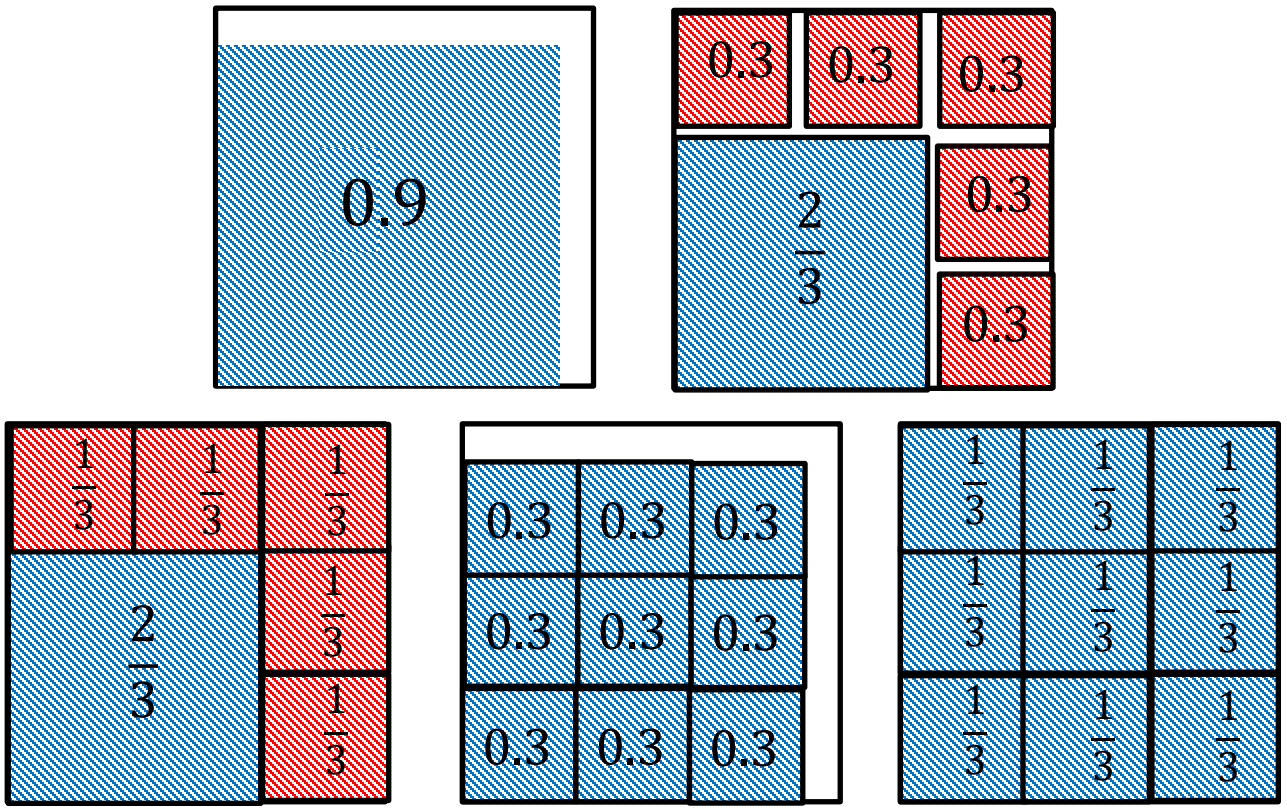


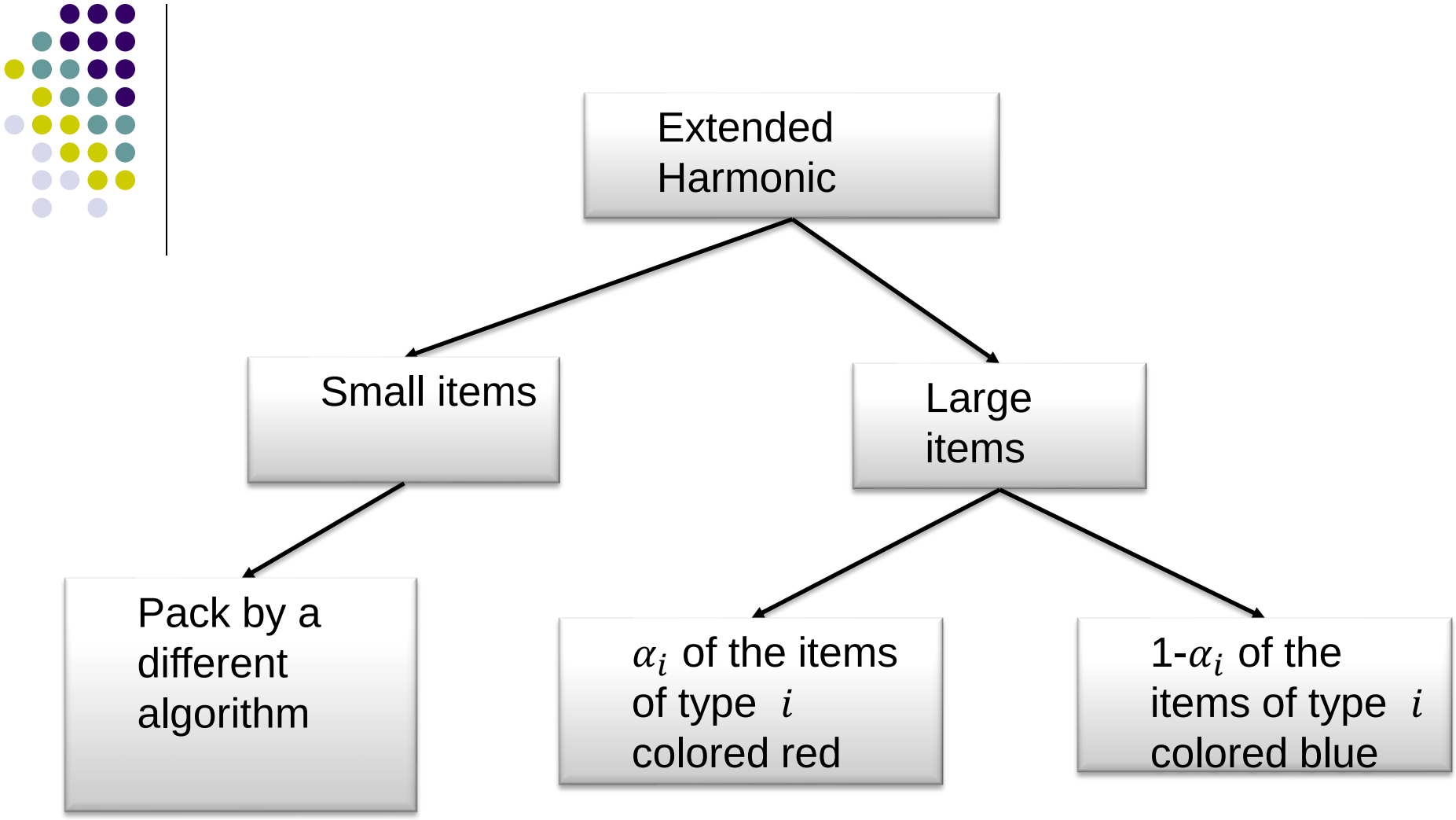
$\alpha_6 = \alpha_5 = 0.4$

Reminder: blue items of type 3 can be packed with red items of type 5,6



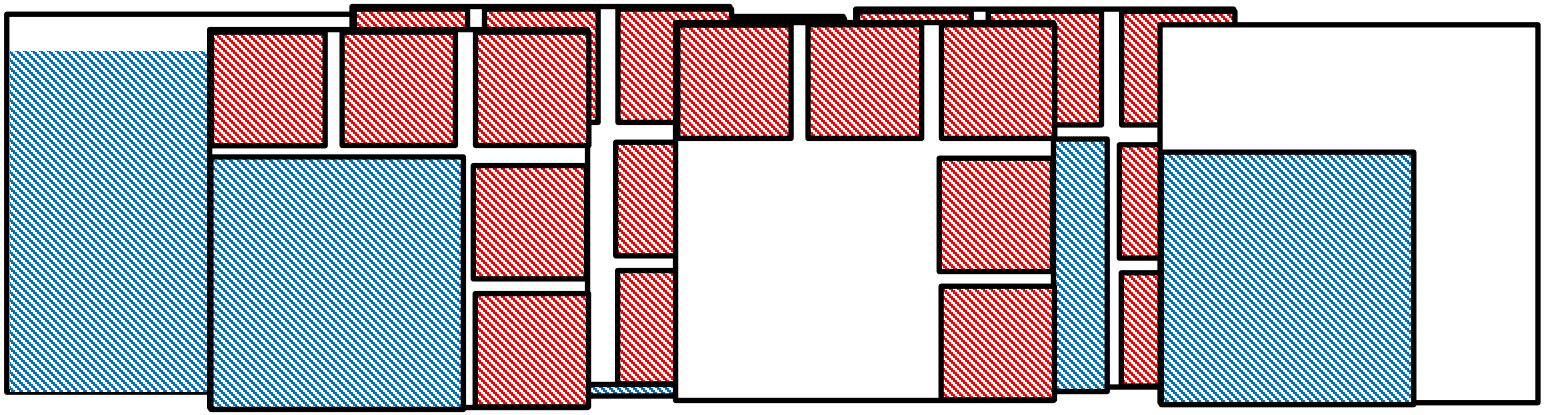
Output


0.9
0.3
0.3
0.3
2/3
0.3
0.3
1/3
1/3
1/3
2/3
1/3
1/3
0.3
0.3
0.3
0.3
0.3
0.3
0.3
0.3
0.3
1/3
1/3
1/3
1/3
1/3
1/3
1/3
1/3
1/3


Extended Harmonic
Small items
Large items
Pack by a different algorithm
α_i of the items of type i colored red
1-α_i of the items of type i colored blue

Extended Harmonic weighting functions

- 3 cases of outputs:
  1. All bins include blue items
  2. All bins that contain blue items which can be packed with red items, are indeed packed with red items
  3. Mixed bins.



Number of bins by weighting function

- Let B_i and R_i be the number of bins containing blue items and red items of type i, respectively.
- Let λ_i be the number of items of type i in the input.
- **Lemma:** $B_i = \frac{1-\alpha_i}{\beta_i^d}\lambda_i + O(1)$, $R_i = \frac{\alpha_i}{\theta_i}\lambda_i + O(1)$

Number of blue items type i that can be contained in one d-dimensional hypercube bin

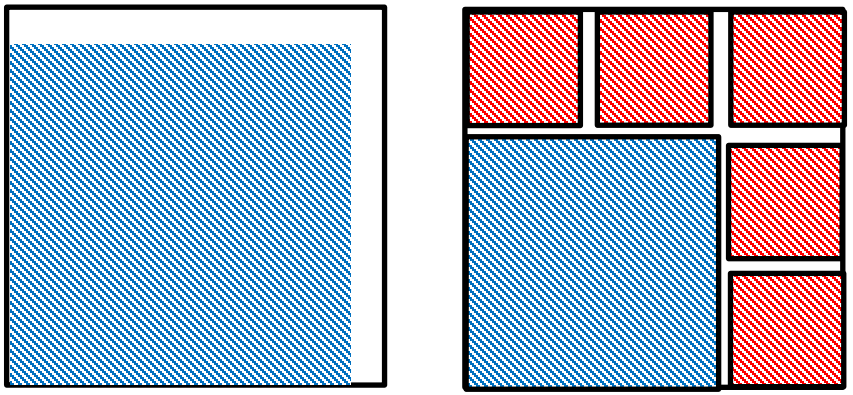
Number of blue items type i in the input.

Number of red items type i that can be contained in one d-dimensional hypercube bin

Number of red items type i in the input.

Case 1

- Every bin in the algorithm's output contain blue item(s) up to a constant number of bins.
- Only blue items "pay" for the bins.



- **Reminder:** $B_i = \frac{1-\alpha_i}{\beta_i^d} \cdot \lambda_i + O(1)$, $R_i = \frac{\alpha_i}{\theta_i} \cdot \lambda_i + O(1)$

Weighting function case 1

$$W_1(p) = \frac{1-\alpha_i}{\beta_i^d}$$

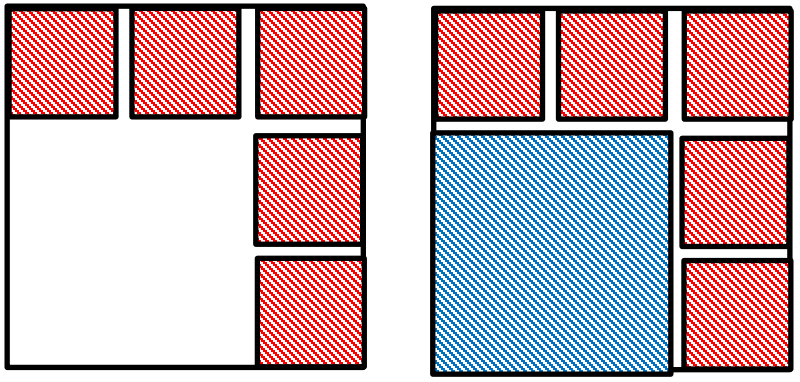
From Lemma we get:

$$A(L) = \sum_i B_i + O(1)$$

Since all the red items in this case are packed in bins that include blue items, the number of bins that include blue items is the number of bins in the algorithm's output in this case.

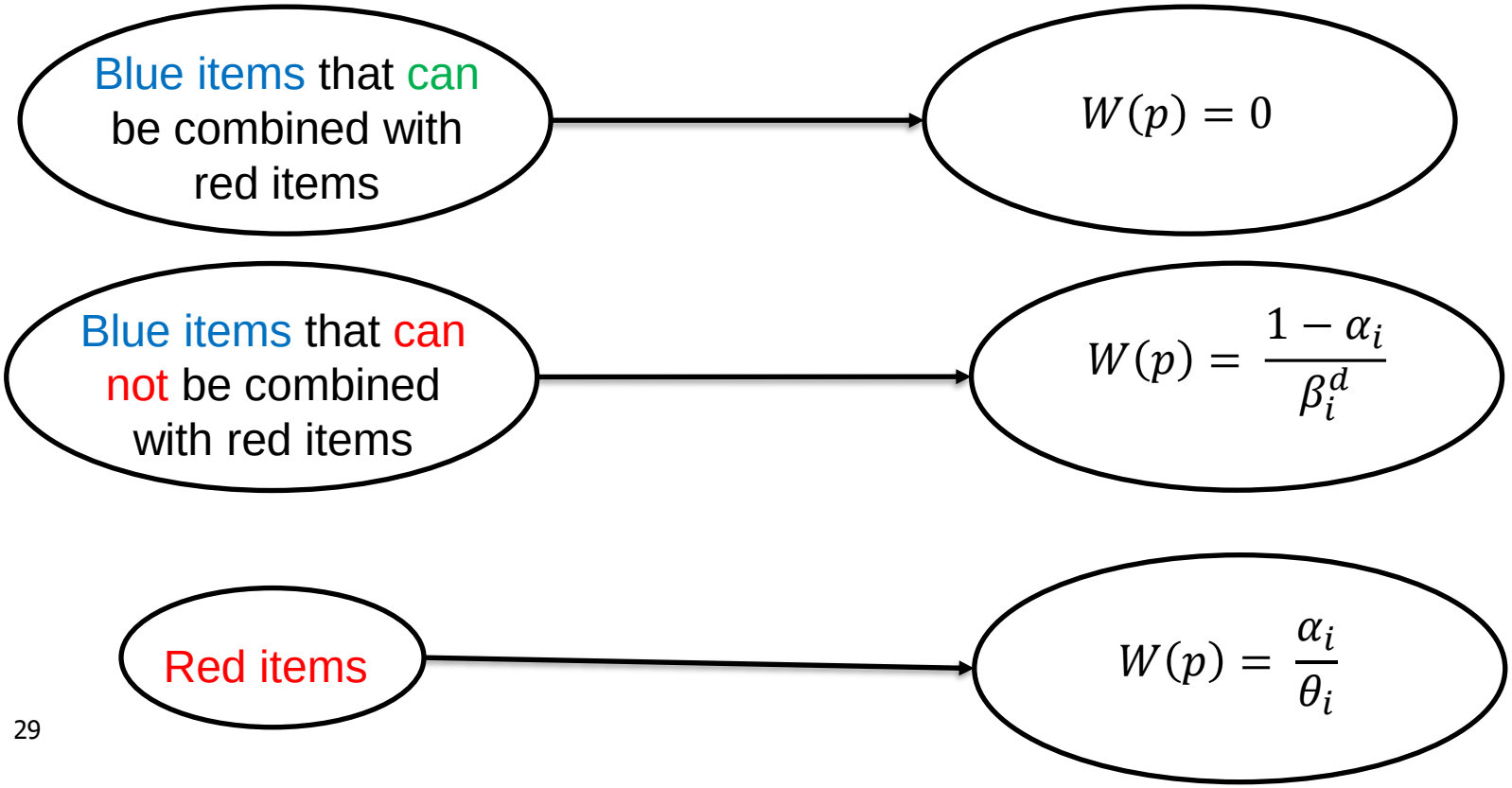
Case 2

- In this case all bins with blue items size above $\frac{1}{3}$ that can be combined with red items were indeed combined.
- Only red items “pay” for these bins.
- Reminder: let θ_i denote the number of red items of type i that can be contained in one bin and let β_i^d denote the number of blue items of type i that can be contained in one bin.



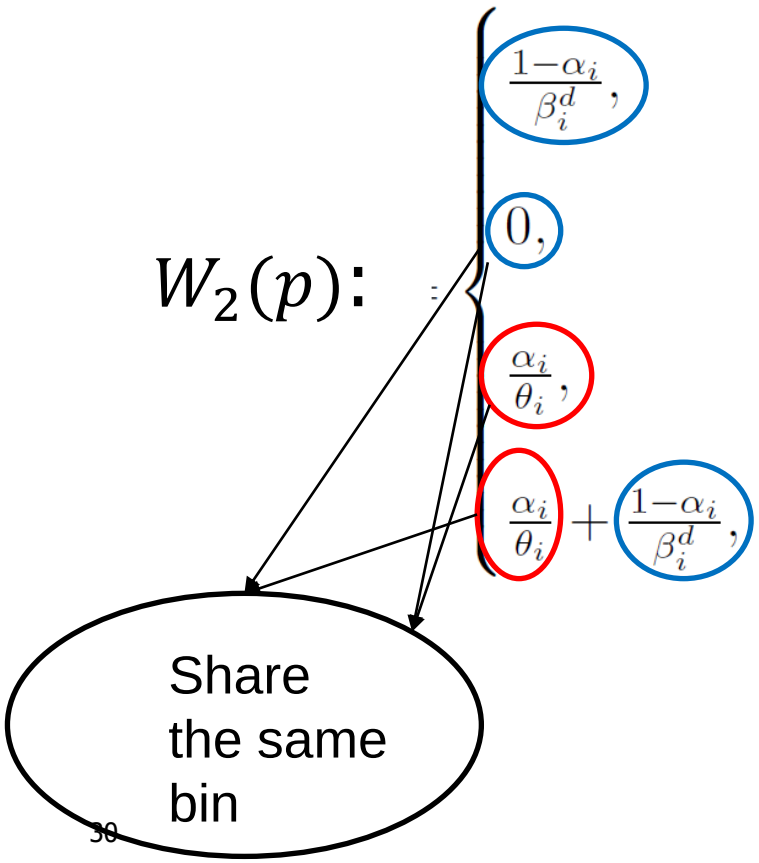
Weighting function of case 2

3 types of items in this case



Weighting function for case 2

In our sample algorithm all blue items of size above $\frac{1}{3}$ can be packed with red items except for blue items in interval $(0.7,1]$, $(0.35,\frac{1}{2}]$ and



$$if\ size(p) \in (0.7,1]$$
$$\text{or} \in (0.4,\tfrac{1}{2}]$$
$$if\ 0.5 < size(p) \leq 0.7$$

Fraction of blue items is 1.

$$if\ \frac{1}{3} < size(p) \leq 0.35$$
$$other$$

Fraction of blue items is $1 - \alpha$.

Reminder: $B_i = \frac{1-\alpha_i}{\beta_i^d} \cdot \lambda_i + O(1)$, $R_i = \frac{\alpha_i}{\theta_i} \cdot \lambda_i + O(1)$

Sketch of proof for case 2

Number of bins in the output:

$$A(L) = \sum_i Bi + \sum_i Ri + \sum_i B_i + R_i + O(1)$$

Items of type i such that all the items are blue ($\alpha_i = 0$) and can not be packed with red items in the same bin

Items of type i such that blue items of this type can be packed with red items, and not all items are colored blue ($\alpha_i \neq 0$)

Items of type i such that blue items of this type can not be packed with red items, and not all items are colored blue ($\alpha_i \neq 0$)

Note that we do not count all the bins that contain blue items of type i such that they can be packed with red items, because we already counted them as red bins, and in this case all bins that contain blue items that can be packed with red items, are indeed packed with them.

- **Reminder:** $B_i = \frac{1-\alpha_i}{\beta_i^d} \cdot \lambda_i + O(1)$, $R_i = \frac{\alpha_i}{\theta_i} \cdot \lambda_i + O(1)$

Summary of case 2

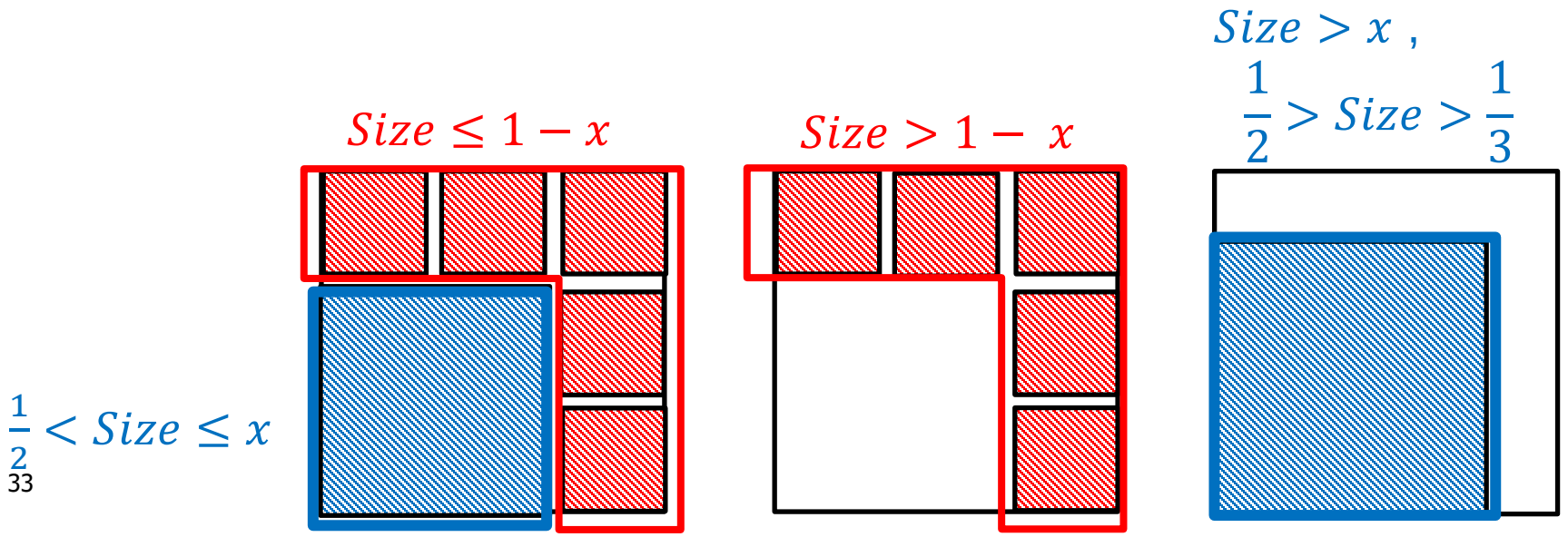
$$W_2(p): \begin{cases} \frac{1-\alpha_i}{\beta_i^d}, & if\ size(p) \in (0.7,1] \text{ or } \in (0.4, \frac{1}{2}] \\ 0, & if\ 0.5 < size(p) \leq 0.7 \\ \frac{\alpha_i}{\theta_i}, & if\ \frac{1}{3} < size(p) \leq 0.35 \\ \frac{\alpha_i}{\theta_i} + \frac{1-\alpha_i}{\beta_i^d}, & other \end{cases} \quad \begin{matrix} \sum_i \mathrm{B}_i \\ \\ \sum_i \mathrm{R}_i \\ \sum_i \mathrm{B}_i + R_i \end{matrix}$$

$$A(L) = \sum_i Bi + \sum_i Ri + \sum_i B_i + R_i$$

Case 3

Note that we have 15 different values of x. Hence, the total number of cases is 17

- In this case all bins with blue items size above x that can be combined with red items were not necessarily combined.
- All bins with blue items size below x and larger than $\frac{1}{2}$ that can be combined with red items were indeed combined.
- All red items with size below $1 - x$ that can be combined with blue items were indeed combined.



$0 \leq w \leq 1$

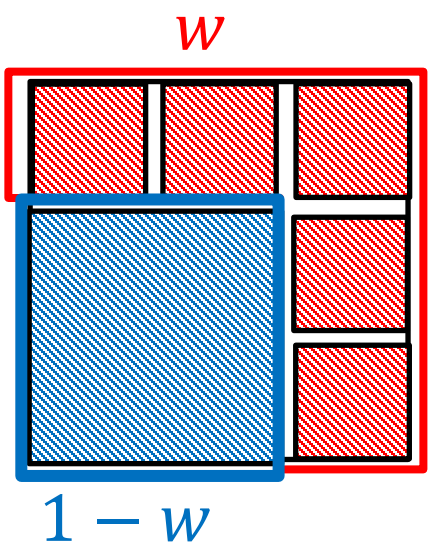
Weighting technique

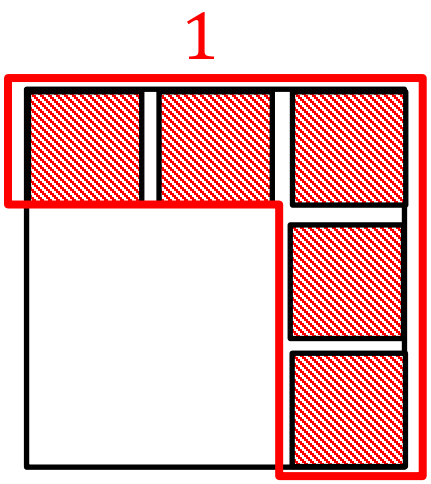
Red items that is combined for sure pay w of the bin

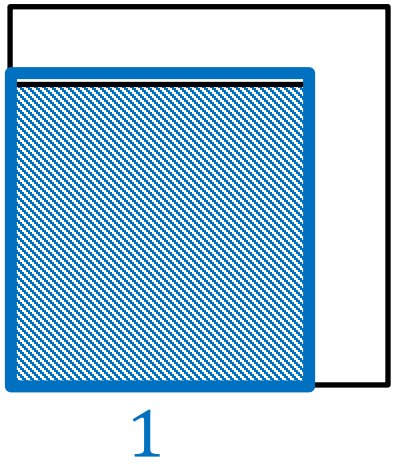
Red items that is not combined for sure pay 1 of the bin

Blue items that is combined for sure pay $1 - w$ of the bin

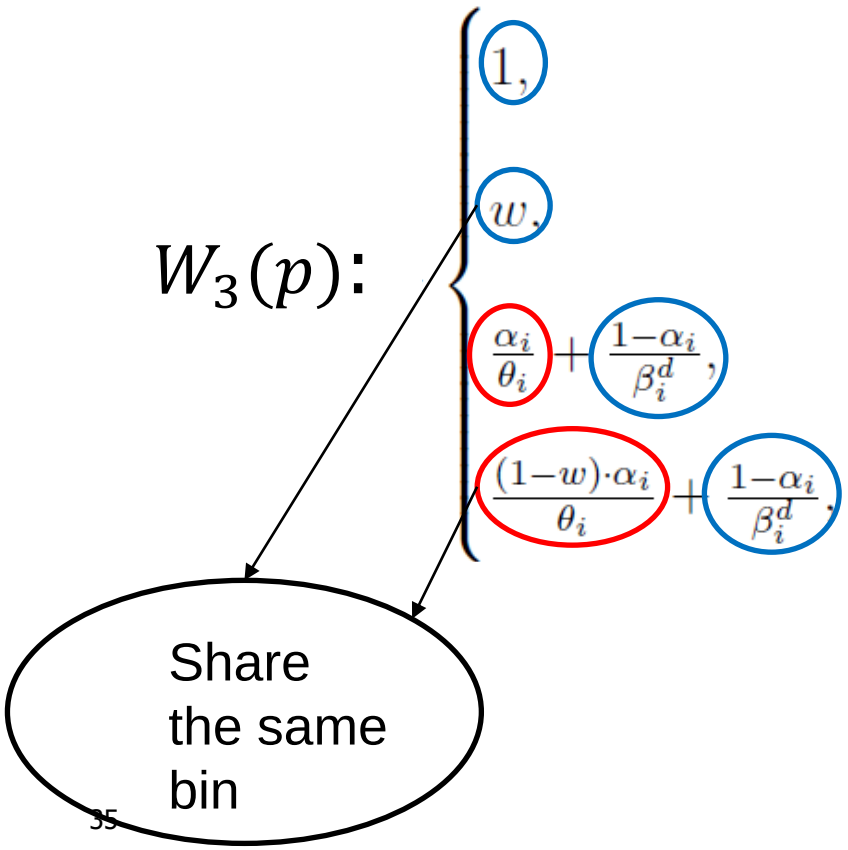
Blue items that is not combined for sure pay 1 of the bin







Weight function for case 3



$if\ size(p) > x$

$if\ 0.5 < size(p) \leq x$

Fraction of blue items is 1.

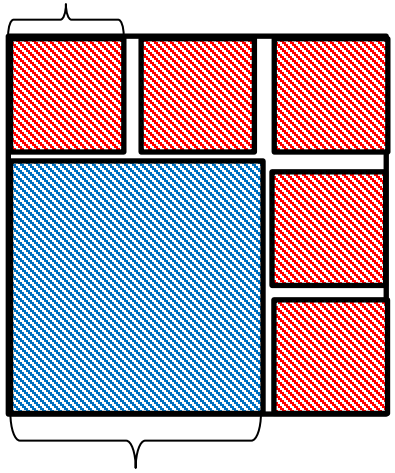
$if\ 1 - x \leq size(p) \leq 0.5$

$if\ \frac{1}{t_{n+1}} < size(p) < 1 - x$

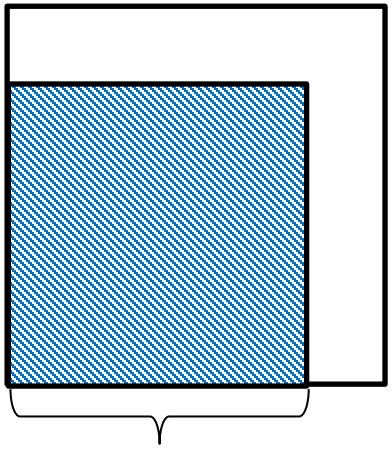
Fraction of blue items is $1 - \alpha$.

Example $x = 0.65$

$Size \leq 0.35$

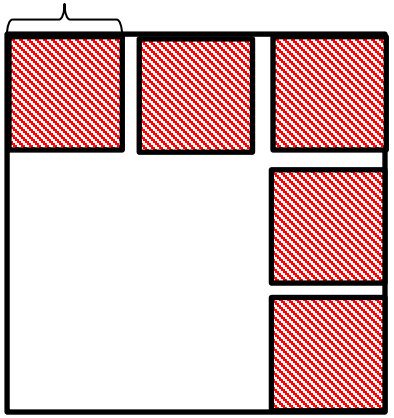


$\frac{1}{2} < Size \leq 0.65$



$Size > 0.65$ or
$Size \leq 0.5$

$Size > 0.35$



- **Reminder:** $B_i = \frac{1-\alpha_i}{\beta_i^d} \cdot \lambda_i + O(1)$, $R_i = \frac{\alpha_i}{\theta_i} \cdot \lambda_i + O(1)$

Sketch of proof for case 3

Number of bins in the output:

$$A(L) = \sum_i Bi + \sum_i 1 - w \cdot Bi + \sum_i B_i + R_i + \sum_i Bi + w \cdot R_i$$

Items of type i such that all the items are blue ($\alpha_i = 0$) and of size $> x$.

Items of sizes between $(0.5, x]$ such that blue items of this type are packed with red items. The fraction of blue items in this type is 1

Items of sizes between $[1 - x. 0.5]$. It is not guaranteed that red items of type i are packed with blue items in the same bin, also it is not guaranteed that blue items of type i are packed with red items.

Items of sizes between $\left(\frac{1}{t_{n+1}}, 1 - x\right]$. Here, red items of these types are packed with blue items, and blue items of these types can not be packed with red items.

- **Reminder:** $B_i = \frac{1-\alpha_i}{\beta_i^d} \cdot \lambda_i + O(1)$, $R_i = \frac{\alpha_i}{\theta_i} \cdot \lambda_i + O(1)$

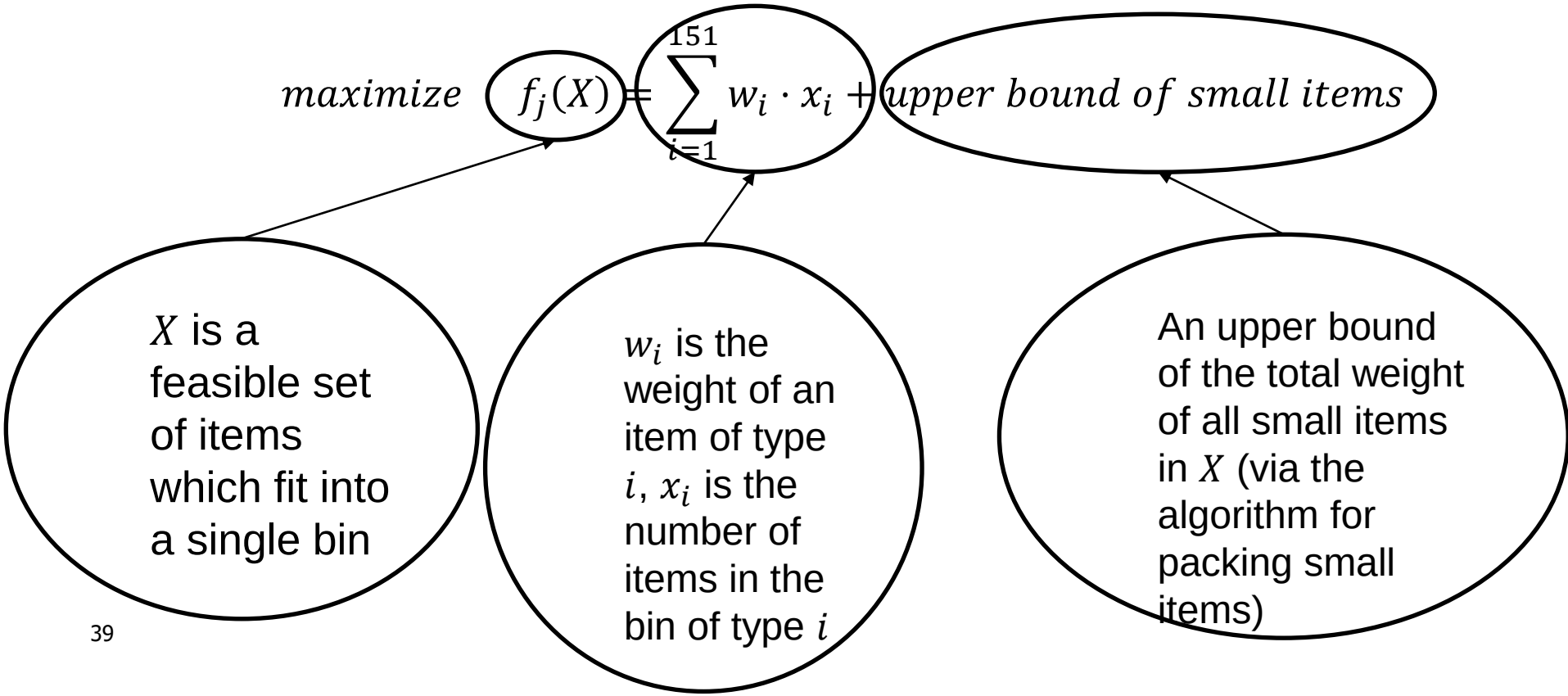
Summary of case 3

$$W_3(p): \begin{cases} 1, & if\ size(p) > x \quad \sum_i B_i \\ w, & if\ 0.5 < size(p) \leq x \quad \sum_i w \cdot B_i \\ \frac{\alpha_i}{\theta_i} + \frac{1-\alpha_i}{\beta_i^d}, & if\ 1-x \leq size(p) \leq 0.5 \quad \sum_i B_i + R_i \\ \frac{(1-w)\cdot\alpha_i}{\theta_i} + \frac{1-\alpha_i}{\beta_i^d}, & if\ \frac{1}{t_{n+1}} < size(p) < 1-x \quad \sum_i B_i + w \cdot R_i \end{cases}$$

$$A(L) = \sum_i Bi + \sum_i w \cdot Bi + \sum_i B_i + R_i + \sum_i Bi + w \cdot R_i$$

Finding the asymptotic competitive ratio

- We use Integer Programing in order to find the competitive ratio.
- For each $j \in \{1, \ldots, \#cases\}$ we find the following:



Constraints

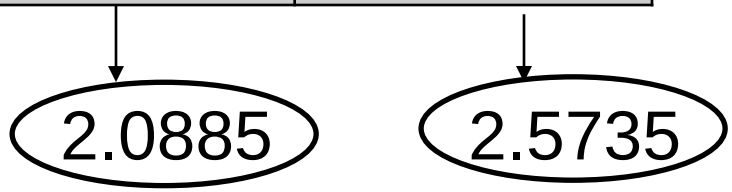
- The previous integer programing is under the following constraints:

$$\sum_{i=1}^{151} x_i \cdot t_{i+1}^{d} \leq 1$$

$$\sum_{i=1}^{151} \left\lfloor \left(t_{i+1} \cdot (u+1)\right) \right\rfloor^{d} \cdot x_i \leq u^{d} \qquad \forall u \in \{1, \ldots, 220\}$$

$$x_i \geq 0 \; and \; x_i \in Z \qquad \forall i \in \{1, \ldots, \#intervals\}$$

Number of case	Square packing	Cube packing
1	2.0884478799685	2.5731896581108
2	1.9438375658626	2.4546421833654
3	2.0109397168059	2.4758230714555
4	1.9607242494316	2.4557193441993
5	1.9942453743436	2.5115525001235
6	1.9875046382360	2.53391757998069
7	1.9554146240072	2.50163026641894
8	1.9441281429162	2.4938219115396
9	2.0884478982863	2.57347626581612
10	2.0884277288254	2.5593413871191126
11	2.0884450773084	2.5567398601707696
12	2.0876840226666	2.557631911023
13	2.0847781920964	2.5498950440578
14	2.0773297796586	2.5226265870712
15	2.0656430335436	2.5277174070986
16	2.0437751234561	2.53854580447380
17	2.0880862874770	2.57186580722798



Questions?