# Lecture 11. Bin Packing

ZHANG Guochuan

December 1, 2021

# Outline

- The First Fit Algorithm
- Karmarkar and Karp's Algorithm

# The First Fit Algorithm

# Problem Statement

## 1-d bin packing

- Input: $n$ items $a_1, a_2, \cdots, a_n$, each wit size $\in (0, 1]$, and an infinite number of unit-size bins
- Output: A feasible packing with the least number of bins used
- feasible packing: all items are packed and no bins accept items summing over 1

# Problem Statement

**1-d bin packing**

- Input: $n$ items $a_1, a_2, \cdots, a_n$, each wit size $\in (0, 1]$, and an infinite number of unit-size bins
- Output: A feasible packing with the least number of bins used
- feasible packing: all items are packed and no bins accept items summing over 1

It is NP-complete to decide if two bins suffice to accommodate given items

# Asymptotic Analysis

- An algorithm $A$ has asymptotic approximation ratio at most $\alpha$ if

$$A(I) \leq \alpha OPT(I) + k$$

  for any instance $I$ and some "constant" $k$.
    - It is absolute approximation ratio when $k = 0$

# Asymptotic Analysis

- An algorithm $A$ has asymptotic approximation ratio at most $\alpha$ if

$$A(I) \leq \alpha OPT(I) + k$$

  for any instance $I$ and some "constant" $k$.
  - It is absolute approximation ratio when $k = 0$
- No algorithm with absolute approximation ratio $< \frac{3}{2}$ unless P=NP

# Asymptotic Analysis

- An algorithm $A$ has asymptotic approximation ratio at most $\alpha$ if

$$A(I) \leq \alpha OPT(I) + k$$

  for any instance $I$ and some "constant" $k$.
  - It is absolute approximation ratio when $k = 0$
- No algorithm with absolute approximation ratio $< \frac{3}{2}$ unless P=NP
- Conventional packing rules: Next-Fit, First-Fit, Best-Fit, Harmonic Fit,...

# Simple Packing Rules

## Next Fit (NL)

- Pack items one by one and keep at a time one bin open
- Pack the current item into the opened bin if it has enough space; otherwise close the bin and open a new one for the item
- $NL(I) \leq 2OPT(I) - 1$

# Simple Packing Rules

## Next Fit (NL)

- Pack items one by one and keep at a time one bin open
- Pack the current item into the opened bin if it has enough space; otherwise close the bin and open a new one for the item
- $NL(I) \leq 2OPT(I) - 1$

## Any Fit

- Pack items one by one and never close a bin before the game is over
- Never open a new bin as long as there exists an opened bin in which the current item fits

# Any Fit

Which bin is chosen if there exist several ones having enough space for the current item?

# Any Fit

Which bin is chosen if there exist several ones having enough space for the current item?

## Specific Fit Rules

- First Fit (FF): choose the earliest one
- Best Fit (BF): choose the fullest one
- Worst Fit (WF): choose the least full one
- Almost Worst Fit (AWF): choose the second least full one

# Any Fit

Which bin is chosen if there exist several ones having enough space for the current item?

## Specific Fit Rules

- First Fit (FF): choose the earliest one
- Best Fit (BF): choose the fullest one
- Worst Fit (WF): choose the least full one
- Almost Worst Fit (AWF): choose the second least full one

## Performance

- WF can be as badly as NF
- The others have an asymptotic ratio of 1.7

## FF

More precisely, we can show that

$$\forall I,\ FF(I) \leq 1.7 OPT(I) + \frac{4}{5}$$

$$I = \{a_1, \cdots, a_n\}, a_i \in (0, 1]$$

$$I = \{a_1, \cdots, a_n\}, a_i \in (0, 1]$$

We aim at

$$\forall I,\ FF(I) \leq 1.7OPT(I) + \frac{4}{5}$$

# Analysis of the FF algorithm

$$I = \{a_1, \cdots, a_n\}, a_i \in (0, 1]$$

We aim at

$$\forall I,\ FF(I) \leq 1.7OPT(I) + \frac{4}{5}$$

Introduce an amortized size (weight): $a_i \to w(a_i)$

# Analysis of the FF algorithm

$$I = \{a_1, \cdots, a_n\}, a_i \in (0,1]$$

We aim at

$$\forall I,\ FF(I) \le 1.7OPT(I) + \frac{4}{5}$$

Introduce an amortized size (weight): $a_i \to w(a_i)$

$$w(I) = \sum_{i=1}^{n} w(a_i) = \sum_{j=1}^{FF(I)} w(B_j) = \sum_{j=1}^{OPT(I)} w(B_j^*)$$

# Analysis of the FF algorithm

$$I = \{a_1, \cdots, a_n\}, a_i \in (0, 1]$$

We aim at

$$\forall I,\ FF(I) \le 1.7OPT(I) + \frac{4}{5}$$

Introduce an amortized size (weight): $a_i \to w(a_i)$

$$w(I) = \sum_{i=1}^{n} w(a_i) = \sum_{j=1}^{FF(I)} w(B_j) = \sum_{j=1}^{OPT(I)} w(B_j^*)$$

- $w(I) \le 1.7OPT(I)$
- $FF(I) \le w(I) + 4/5$

# Analysis of the FF algorithm

$$I = \{a_1, \cdots, a_n\}, a_i \in (0, 1]$$

We aim at

$$\forall I,\ FF(I) \leq 1.7OPT(I) + \frac{4}{5}$$

Introduce an amortized size (weight): $a_i \to w(a_i)$

$$w(I) = \sum_{i=1}^{n} w(a_i) = \sum_{j=1}^{FF(I)} w(B_j) = \sum_{j=1}^{OPT(I)} w(B_j^*)$$

- $w(I) \leq 1.7OPT(I) \qquad \Longleftarrow \quad \forall B_j^*, w(B_j^*) \leq 1.7$
- $FF(I) \leq w(I) + 4/5$

# Analysis of the FF algorithm

$$I = \{a_1, \cdots, a_n\}, a_i \in (0, 1]$$

We aim at

$$\forall I,\ FF(I) \leq 1.7OPT(I) + \frac{4}{5}$$

Introduce an amortized size (weight): $a_i \to w(a_i)$

$$w(I) = \sum_{i=1}^{n} w(a_i) = \sum_{j=1}^{FF(I)} w(B_j) = \sum_{j=1}^{OPT(I)} w(B_j^*)$$

- $w(I) \leq 1.7OPT(I) \quad \Longleftarrow \quad \forall B_j^*, w(B_j^*) \leq 1.7$
- $FF(I) \leq w(I) + 4/5 \quad \Longleftarrow \quad \forall B_j, w(B_j) \geq 1$ in an "average" way

# Weighting Function

$$w(a) = \frac{6}{5}a + v(a)$$

# Weighting Function

$$w(a) = \frac{6}{5}a + v(a)$$

$\frac{6}{5}a$: called "scaled size", $v(a)$: bonus

# Weighting Function

$$w(a) = \frac{6}{5}a + v(a)$$

$\frac{6}{5}a$: called "scaled size", $v(a)$: bonus

$$v(a) = \begin{cases} 0 & \text{if } a \leq \frac{1}{6} \\ \frac{3}{5}(a - \frac{1}{6}) & \text{if } \frac{1}{6} \leq a \leq \frac{1}{3} \\ \frac{1}{10} & \text{if } \frac{1}{3} \leq a \leq \frac{1}{2} \\ \frac{4}{10} & \text{if } a > \frac{1}{2} \end{cases}$$

We call the items, respectively, tiny, small, medium and big based on the above classification

# Weights of Optimal Bins

$$w(a) = \frac{6}{5}a + v(a)$$

$$v(a) = \begin{cases} 0 & \text{if } a \le \frac{1}{6} \\ \frac{3}{5}(a - \frac{1}{6}) & \text{if } \frac{1}{6} \le a \le \frac{1}{3} \\ \frac{1}{10} & \text{if } \frac{1}{3} \le a \le \frac{1}{2} \\ \frac{4}{10} & \text{if } a > \frac{1}{2} \end{cases}$$

$\forall B_j^*, w(B_j^*) \le 1.7$

# Weights of Optimal Bins

$$w(a) = \frac{6}{5}a + v(a)$$

$$v(a) = \begin{cases} 0 & \text{if } a \le \frac{1}{6} \\ \frac{3}{5}(a - \frac{1}{6}) & \text{if } \frac{1}{6} \le a \le \frac{1}{3} \\ \frac{1}{10} & \text{if } \frac{1}{3} \le a \le \frac{1}{2} \\ \frac{4}{10} & \text{if } a > \frac{1}{2} \end{cases}$$

$\forall B_j^*, w(B_j^*) \le 1.7$

- If $B_j^*$ does not contain any big item, $w(B_j^*) \le 1.2 + 0.3 < 1.7$

# Weights of Optimal Bins

$$w(a) = \frac{6}{5}a + v(a)$$

$$v(a) = \begin{cases} 0 & \text{if } a \le \frac{1}{6} \\ \frac{3}{5}(a - \frac{1}{6}) & \text{if } \frac{1}{6} \le a \le \frac{1}{3} \\ \frac{1}{10} & \text{if } \frac{1}{3} \le a \le \frac{1}{2} \\ \frac{4}{10} & \text{if } a > \frac{1}{2} \end{cases}$$

$\forall B_j^*, w(B_j^*) \le 1.7$

- If $B_j^*$ does not contain any big item, $w(B_j^*) \le 1.2 + 0.3 < 1.7$
- If $B_j^*$ contains a big item, $w(B_j^*) \le 1.2 + 0.4 + 0.1 = 1.7$

# Weights of FF Bins

$$w(a) = \frac{6}{5}a + v(a)$$

$$v(a) = \begin{cases} 0 & \text{if } a \leq \frac{1}{6} \\ \frac{3}{5}(a - \frac{1}{6}) & \text{if } \frac{1}{6} \leq a \leq \frac{1}{3} \\ \frac{1}{10} & \text{if } \frac{1}{3} \leq a \leq \frac{1}{2} \\ \frac{4}{10} & \text{if } a > \frac{1}{2} \end{cases}$$

$\forall\ B_j, w(B_j) \geq 1$ in an average way

# Weights of FF Bins

$$w(a) = \frac{6}{5}a + v(a)$$

$$v(a) = \begin{cases} 0 & \text{if } a \le \frac{1}{6} \\ \frac{3}{5}(a - \frac{1}{6}) & \text{if } \frac{1}{6} \le a \le \frac{1}{3} \\ \frac{1}{10} & \text{if } \frac{1}{3} \le a \le \frac{1}{2} \\ \frac{4}{10} & \text{if } a > \frac{1}{2} \end{cases}$$

$\forall\ B_j, w(B_j) \ge 1$ in an average way

- If $B_j$ has a content $c(B_j) \ge \frac{5}{6}$, $w(B_j) \ge 1$

# Weights of FF Bins

$$w(a) = \frac{6}{5}a + v(a)$$

$$v(a) = \begin{cases} 0 & \text{if } a \le \frac{1}{6} \\ \frac{3}{5}(a - \frac{1}{6}) & \text{if } \frac{1}{6} \le a \le \frac{1}{3} \\ \frac{1}{10} & \text{if } \frac{1}{3} \le a \le \frac{1}{2} \\ \frac{4}{10} & \text{if } a > \frac{1}{2} \end{cases}$$

$\forall\ B_j, w(B_j) \ge 1$ in an average way

- If $B_j$ has a content $c(B_j) \ge \frac{5}{6}$, $w(B_j) \ge 1$
- If $B_j$ contains a big item, $w(B_j) > 1.2 \times 0.5 + 0.4 = 1$

# Weights of FF Bins

$$w(a) = \frac{6}{5}a + v(a)$$

$$v(a) = \begin{cases} 0 & \text{if } a \le \frac{1}{6} \\ \frac{3}{5}(a - \frac{1}{6}) & \text{if } \frac{1}{6} \le a \le \frac{1}{3} \\ \frac{1}{10} & \text{if } \frac{1}{3} \le a \le \frac{1}{2} \\ \frac{4}{10} & \text{if } a > \frac{1}{2} \end{cases}$$

$\forall\ B_j, w(B_j) \ge 1$ in an average way

- If $B_j$ has a content $c(B_j) \ge \frac{5}{6}$, $w(B_j) \ge 1$
- If $B_j$ contains a big item, $w(B_j) > 1.2 \times 0.5 + 0.4 = 1$
- If $B_j$ contains two medium items, $w(B_j) \ge \frac{6}{5} \cdot \frac{2}{3} + 0.2 = 1$

# Weights of FF Bins

$$w(a) = \frac{6}{5}a + v(a)$$

$$v(a) = \begin{cases} 0 & \text{if } a \le \frac{1}{6} \\ \frac{3}{5}(a - \frac{1}{6}) & \text{if } \frac{1}{6} \le a \le \frac{1}{3} \\ \frac{1}{10} & \text{if } \frac{1}{3} \le a \le \frac{1}{2} \\ \frac{4}{10} & \text{if } a > \frac{1}{2} \end{cases}$$

Sort the FF bins in the order that they appear. Remove all those bins having weights at least one. Now consider the left bins (keeping the order). Note that each of such bins

- does not contain a large item;
- contains at most one medium item;
- has a content less than $\frac{5}{6}$

# Further Observations

Let $k$ be the number of bins left. Without loss of generality, assume $k \geq 2$

- ♣ Only one item in a bin
- ♣ At least two items in a bin

# Further Observations

Let $k$ be the number of bins left. Without loss of generality, assume $k \geq 2$

♣ Only one item in a bin

- The item cannot be a big one

♣ At least two items in a bin

# Further Observations

Let $k$ be the number of bins left. Without loss of generality, assume $k \geq 2$

♣ Only one item in a bin

- The item cannot be a big one
- If exists, it must be the last bin $B_p$

♣ At least two items in a bin

## Further Observations

Let $k$ be the number of bins left. Without loss of generality, assume $k \geq 2$

♣ Only one item in a bin

- The item cannot be a big one
- If exists, it must be the last bin $B_p$

♣ At least two items in a bin

- All but at most one have a content larger than 2/3;

# Further Observations

Let $k$ be the number of bins left. Without loss of generality, assume $k \geq 2$

♣ Only one item in a bin

- The item cannot be a big one
- If exists, it must be the last bin $B_p$

♣ At least two items in a bin

- All but at most one have a content larger than 2/3;
- If there exists a bin whose content is less than 2/3, it must be either the last (if $B_p$ do not exist) or the second last bin $B_q$ (if $B_p$ exists)

# Analysis

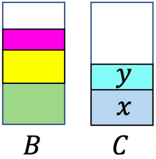
Put $B_p$ aside. Consider any two adjacent bins $B$ and $C$:

- Recall that $2/3 \leq c(B) < 5/6$. Suppose $c(B) = 5/6 - z, z \in (0, 1/6]$

# Analysis

Put $B_p$ aside. Consider any two adjacent bins $B$ and $C$:

- Recall that $2/3 \leq c(B) < 5/6$. Suppose $c(B) = 5/6 - z, z \in (0, 1/6]$

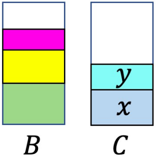



$$x > 1 - (\tfrac{5}{6} - z) = \tfrac{1}{6} + z \in (\tfrac{1}{6}, \tfrac{1}{3}], \;\; y \in (\tfrac{1}{6}, \tfrac{1}{3}]$$

# Analysis

Put $B_p$ aside. Consider any two adjacent bins $B$ and $C$:

- Recall that $2/3 \leq c(B) < 5/6$. Suppose $c(B) = 5/6 - z, z \in (0, 1/6]$




$$x > 1 - (\tfrac{5}{6} - z) = \tfrac{1}{6} + z \in (\tfrac{1}{6}, \tfrac{1}{3}], \;\; y \in (\tfrac{1}{6}, \tfrac{1}{3}]$$

$$v(x) > \tfrac{3}{5}(\tfrac{1}{6} + z - \tfrac{1}{6}) = \tfrac{3}{5}z, \;\; v(y) > \tfrac{3}{5}z$$

- $h = \frac{6}{5}c(B) + v(C) = \frac{6}{5}(\frac{5}{6} - z) + v(x) + v(y)$

$$\geq \tfrac{6}{5}(\tfrac{5}{6} - z) + \tfrac{3}{5}z + \tfrac{3}{5}z = 1$$

# Analysis

No matter if $B_p$ and $B_q$ exist, we apply the above analysis from the first bin to the second last bin.

- For $i = 1, 2, \ldots, k-2$, $\frac{6}{5}c(B_i) + v(B_{i+1}) \geq 1$
- $c(B_{k-1}) + c(B_k) > 1$
- $$\begin{aligned} \sum_{i=1}^{k} w(B_i) &\geq \sum_{i=1}^{k-2}(\frac{6}{5}c(B_i) + v(B_{i+1})) + \frac{6}{5}(c(B_{k-1}) + c(B_k)) \\ &\geq (k-2) + \frac{6}{5} = k - \frac{4}{5} \end{aligned}$$

# Analysis

At this moment, we have shown

**Summary**

- Each optimum bin has a weight at most 1.7
- With a total loss of 4/5, the average weight of FF bins is at least one

which implies that

- $W(I) \leq 1.7OPT(I)$
- $W(I) \geq FF(I) - 4/5$

So, $FF(I) \leq 1.7OPT(I) + \frac{4}{5}$

# Analysis

At this moment, we have shown

**Summary**

- Each optimum bin has a weight at most 1.7
- With a total loss of 4/5, the average weight of FF bins is at least one

which implies that

- $W(I) \leq 1.7OPT(I)$
- $W(I) \geq FF(I) - 4/5$

So, $FF(I) \leq 1.7OPT(I) + \frac{4}{5}$

A tight analysis shows $FF(I) \leq 1.7OPT(I)$ (Dosa and Sgall 2013)

# Better Results

- First Fit Decreasing (FFD): apply First Fit after sorting the items in non-increasing order of their sizes
- $FFD(I) \leq \frac{3}{2}OPT(I)$ (best possible assuming that $P \neq NP$)
- $FFD(I) \leq \frac{11}{9}OPT(I) + \frac{6}{9}$ (Dosa 2007)

# Even Better Results

Bin packing can be approximated to

- $OPT + O(\log^2 OPT)$ (Karmarkar and Karp 1982)
- $OPT + O(\log OPT * \log \log OPT)$ (Rothvoss 2013)
- $OPT + O(\log OPT)$ (Rothvoss and Hoberg 2017)

# Karmarkar and Karp's Algorithm

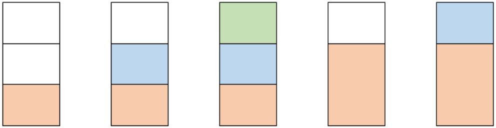
# Packing Configurations

- $I = \{a_1, a_2, \ldots, a_n\}$
- $m$ : the number of different item sizes
- $b_i$ : the number of items of size $s_i$

# Packing Configurations

- $I = \{a_1, a_2, \dots, a_n\}$
- $m$ : the number of different item sizes
- $b_i$ : the number of items of size $s_i$

Configuration: a feasible packing into a bin



# Configuration LP

$N$ : the number of different configurations
$t_{ij}$ : the number of items of size $s_i$ in configuration $j$

# Configuration LP

$N$ : the number of different configurations
$t_{ij}$ : the number of items of size $s_i$ in configuration $j$

$$\begin{aligned} \min \quad & \sum_{j=1}^{N} x_j & & \text{(Configuration LP)} \\ \text{s.t.} \quad & \sum_{j=1}^{N} t_{ij} x_j \geq b_i & i = 1, \ldots, m \\ & x_j \geq 0 & j = 1, \ldots, N \end{aligned}$$

# Configuration LP

**Theorem**

The configuration LP can be solved within an additive error of at most 1 in time polynomial in $m$ and $\log(n/s_m)$, where $s_m$ is the size of smallest item in the instance.

# Configuration LP

**Theorem**

The configuration LP can be solved within an additive error of at most 1 in time polynomial in $m$ and $\log(n/s_m)$, where $s_m$ is the size of smallest item in the instance.

- Ensure that $s_m \geq 1/SIZE(I)$, where $SIZE(I) = \sum s_i$.
- Those items smaller than $1/SIZE(I)$ can be packed by FF or NF without increasing the additive error much.
- Therefore, the configuration LP can be solved in polynomial time.

# Rounding the LP Solution

## Rounding Scheme

There are at most $m$ non-zero variables in an extreme point. Directly rounding up can only guarantee a feasible packing with $OPT(I) + m$ bins. To do better we round down the optimal LP solution $x^*$:

- Pack $\lfloor x_j^* \rfloor$ bins according to the configuration, for $j = 1, 2, \ldots, m$.
- Denote the set of items already packed as $I_z$.
- $SIZE(I - I_z) \leq m$.
- Recurse on the remaining items $I - I_z$

## Claim

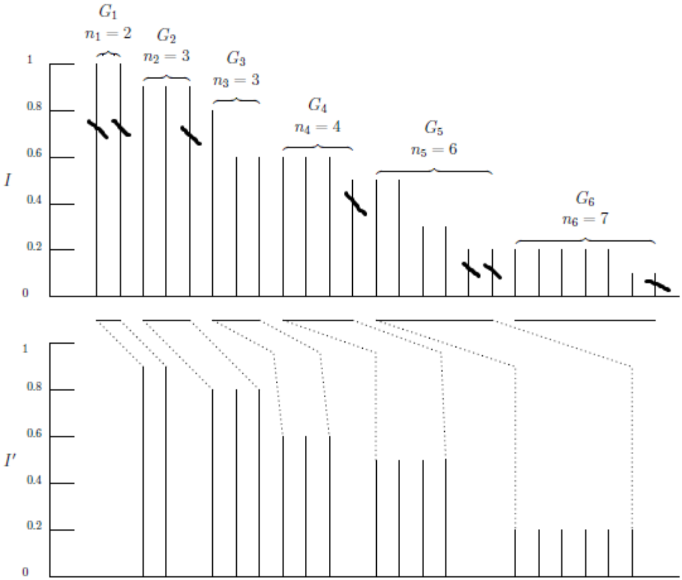
$LP(I - I_z) + LP(I_z) \leq LP(I)$.

Proof. $LP(I - I_z) + LP(I_z) \leq \sum_j (x_j^* - \lfloor x_j^* \rfloor) + \sum_j \lfloor x_j^* \rfloor = LP(I)$.

# Harmonic Grouping

- Sort the items in non-increasing order.
- Following the order we form a group whenever the total size is at least 2, and start a new group with the next item.
- Let $r$ be the number of groups, where $G_i$ is the $i$-th group with $n_i$ items, for $i = 1, 2, \ldots, r$.
- $n_i \geq n_{i-1}$, $i = 2, \ldots, r-1$
- Discard group $G_1$ and $G_r$.
- For $i = 2, \ldots, r-1$, discard $n_i - n_{i-1}$ smallest items in $G_i$, and round the remaining $n_{i-1}$ items to the size of the largest item in $G_i$.
- The remaining items form a new instance $I'$.
- $LP(I) \geq LP(I')$.

# Illustration of the Grouping



# Key Issues

- $I \rightarrow I'$ by harmonic grouping;
- The number $m$ of distinct item sizes in $I'$ is at most $SIZE(I)/2$;
- The total size of all discarded items is $O(\log SIZE(I))$.

# Algorithm

BIN PACK($I$)

$k = 1$

**if** $SIZE(I) < 10$ **then**

  Pack remaining items using First Fit

**else**

  Apply **harmonic grouping scheme** to create instance $I'$; pack discarded items in $O(\log SIZE(I))$ bins using First Fit

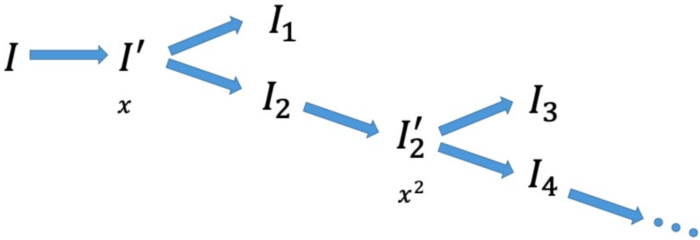
  Let $x$ be optimal solution to configuration LP for instance $I'$

  Pack $\lfloor x_j \rfloor$ bins in configuration $T_j$ for $j = 1, \cdots, N$; call the packed items instance $I_{2k-1}$

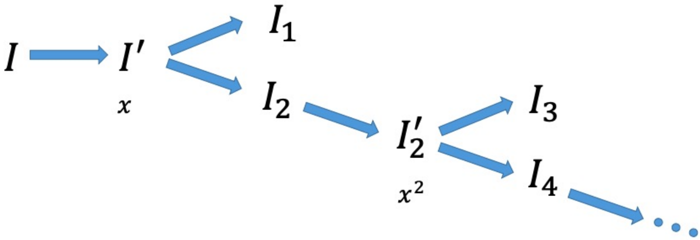
  Let $I_{2k}$ be remaining items from $I'$

  Pack $I_{2k}$ via BIN PACK($I_{2k}$); $k = k + 1$

# Algorithm



# Algorithm




The harmonic grouping scheme can be used to design an approximation algorithm that always finds a packing with $OPT_{LP}(I) + O(\log^2(SIZE(I)))$ bins.

# Analysis

**All the discarded items can be packed in** $O(\log^2(SIZE(I)))$ **bins.**

# Analysis

**All the discarded items can be packed in** $O(\log^2(SIZE(I)))$ **bins.**

**The sum of the integral parts (**$I_1, I_3, \cdots$**) is at most** $OPT_{LP}(I)$.

# Analysis

**All the discarded items can be packed in $O(\log^2(SIZE(I)))$ bins.**

- $I \to I'$ by harmonic grouping;
- The number of distinct item sizes in $I'$ is at most $SIZE(I)/2$;
- The total size of all discarded items is $O(\log SIZE(I))$;

**The sum of the integral parts ($I_1, I_3, \cdots$) is at most $OPT_{LP}(I)$.**

# Analysis

**All the discarded items can be packed in $O(\log^2(SIZE(I)))$ bins.**

- $I \rightarrow I'$ by harmonic grouping;
- The number of distinct item sizes in $I'$ is at most $SIZE(I)/2$;
- The total size of all discarded items is $O(\log SIZE(I))$;

**The sum of the integral parts ($I_1, I_3, \cdots$) is at most $OPT_{LP}(I)$.**

- $OPT_{LP}(I_1) + OPT_{LP}(I_2) \leq OPT_{LP}(I') \leq OPT_{LP}(I)$;

# Bounding the Discarded Items

- $I \to I'$ by harmonic grouping;
- The number of distinct item sizes in $I'$ is at most $SIZE(I)/2$;
  (Each of these groups has size at least 2)
- The total size of all discarded items is $O(\log SIZE(I))$;
  (Each of these groups has size at most 3)
  $n_1 \leq n_2 \leq \cdots n_{r-1}$
  $r \leq \lceil \frac{SIZE(I)}{2} \rceil$
  $SIZE(G_1) + SIZE(G_r) \leq 6$

# Bounding the Discarded Items

- $I \to I'$ by harmonic grouping;
- The number of distinct item sizes in $I'$ is at most $SIZE(I)/2$;
  (Each of these groups has size at least 2)
- The total size of all discarded items is $O(\log SIZE(I))$;
  (Each of these groups has size at most 3)
  $n_1 \leq n_2 \leq \cdots n_{r-1}$
  $r \leq \lceil \frac{SIZE(I)}{2} \rceil$
  $SIZE(G_1) + SIZE(G_r) \leq 6$
  Average size of the discarded items in group $G_i$ is at most $3/n_i$

# Bounding the Discarded Items

- $I \to I'$ by harmonic grouping;
- The number of distinct item sizes in $I'$ is at most $SIZE(I)/2$;
  (Each of these groups has size at least 2)
- The total size of all discarded items is $O(\log SIZE(I))$;
  (Each of these groups has size at most 3)
  $n_1 \leq n_2 \leq \cdots n_{r-1}$
  $r \leq \lceil \frac{SIZE(I)}{2} \rceil$
  $SIZE(G_1) + SIZE(G_r) \leq 6$
  Average size of the discarded items in group $G_i$ is at most $3/n_i$

  $SIZE(I_d) = SIZE(G_1) + SIZE(G_r) + \sum_{i=2}^{r-1}(n_i - n_{i-1})\frac{3}{n_i} \leq 6 + \sum_{i=1}^{r-1} \frac{3}{i} = O(\log SIZE(I))$