Online Bin Packing

Shahin Kamali

Lecture 16 - Nov. 2nd, 2017
University of Manitoba
Review & Plan
Today’s objectives

- Competitive ratio of Next Fit (Review)
- Competitive ratio of Harmonic
  - Weighting technique for upper bound
  - Lower bound!
- A sketch of competitive ratio of Best Fit and First Fit
Competitive Analysis of Next Fit

- Competitive ratio of NextFit is at most 2.
Competitive Analysis of Next Fit

- Competitive ratio of NextFit is at most 2.
  - In the final packing, total size of items in each two consecutive bins is larger than 1.
  - The cost of NextFit for serving $\sigma$ is smaller than $2S(\sigma)$ where $S(\sigma)$ is the total size of items in $\sigma$.
    - Assume $\text{cost}(\text{NextFit}) = k$
    - Each two consecutive bins have total size $> 1 \rightarrow$ total-size $S(\sigma)$ of items in $\sigma$ is more than $k/2$
Competitive Analysis of Next Fit

- Competitive ratio of NextFit is at most 2.
  - In the final packing, total size of items in each two consecutive bins is larger than 1.
  - The cost of NextFit for serving $\sigma$ is smaller than $2S(\sigma)$ where $S(\sigma)$ is the total size of items in $\sigma$.
    - Assume $\text{cost(NextFit)} = k$
    - Each two consecutive bins have total size $> 1 \rightarrow$ total-size $S(\sigma)$ of items in $\sigma$ is more than $k/2$
  - $\text{OPT}(\sigma) \geq S(\sigma)$: Even when $\text{OPT}$ packs items tightly (with no wasted space), $S(\sigma)$ bins are required.
Competitive Analysis of Next Fit

- Competitive ratio of NextFit is at most 2.
  - In the final packing, total size of items in each two consecutive bins is larger than 1.
  - The cost of NextFit for serving $\sigma$ is smaller than $2S(\sigma)$ where $S(\sigma)$ is the total size of items in $\sigma$.
    - Assume $\text{cost}(\text{NextFit}) = k$
    - Each two consecutive bins have total size $> 1 \rightarrow$ total-size $S(\sigma)$ of items in $\sigma$ is more than $k/2$
  - $\text{OPT}(\sigma) \geq S(\sigma)$: Even when $\text{OPT}$ packs items tightly (with no wasted space), $S(\sigma)$ bins are required.
Competitive Analysis of Next Fit

- Competitive ratio of NextFit is at least 2.
Competitive Analysis of Next Fit

- Competitive ratio of NextFit is at least 2.
  - Consider sequence $\sigma = \langle 0.5, \epsilon, 0.5, \epsilon, \ldots \rangle$. 
Competitive Analysis of Next Fit

- Competitive ratio of NextFit is at least 2.
  - Consider sequence $\sigma = <0.5, \epsilon, 0.5, \epsilon, \ldots>$.
  - The cost of NextFit for serving $\sigma$ is roughly $n/2$
    ($n$ is the length of $\sigma$).
  - The cost of $\text{OPT}$ is roughly $n/4$. 

Theorem

Competitive ratio of NextFit is exactly 2.
Competitive Analysis of Next Fit

- Competitive ratio of NextFit is at least 2.
  - Consider sequence $\sigma = \langle 0.5, \epsilon, 0.5, \epsilon, \ldots \rangle$.
  - The cost of NextFit for serving $\sigma$ is roughly $n/2$ ($n$ is the length of $\sigma$).
  - The cost of $\text{OPT}$ is roughly $n/4$.
Competitive Analysis of Next Fit

- Competitive ratio of NextFit is at least 2.
  - Consider sequence $\sigma = \langle 0.5, \epsilon, 0.5, \epsilon, \ldots \rangle$.
  - The cost of NextFit for serving $\sigma$ is roughly $n/2$ ($n$ is the length of $\sigma$).
  - The cost of $\text{OPT}$ is roughly $n/4$.

\[ \begin{array}{cccccccc}
\epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\
5 & 5 & 5 & 5 & 5 & 5 \\
\end{array} \quad \begin{array}{cccc}
5 & 5 & 5 & \\
5 & 5 & 5 \\
\end{array} \]

NextFit                        OPT

Theorem

*Competitive ratio of NextFit is exactly 2.*
Weighting Argument for Harmonic Algorithm
Harmonic Algorithm classes: \((\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots, (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}]\).

Place members of each class separately from others.
Harmonic Algorithm

Harmonic Algorithm classes: \((\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots, (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}]\).

Place members of each class separately from others.

\[ \text{Harmonic} \quad K = 4 \]

\[ < 0.9 \ 0.3 \ 0.8 \ 0.5 \ 0.1 \ 0.1 \ 0.3 \ 0.2 \ 0.4 \ 0.2 \ 0.4 \ 0.5 \ 0.5 \ 0.8 \ 0.6 \ 0.4 \ 0.5 \ldots > \]
Harmonic Algorithm classes: \((\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots, (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}]\).

Place members of each class separately from others.

Harmonic  \(K = 4\)

\[
< 0.9 \ 0.3 \ 0.8 \ 0.5 \ 0.1 \ 0.1 \ 0.3 \ 0.2 \ 0.4 \ 0.2 \ 0.4 \ 0.5 \ 0.5 \ 0.8 \ 0.6 \ 0.4 \ 0.5 \ldots >
\]

\[
x > \frac{1}{2} \quad \frac{1}{3} < x \leq \frac{1}{2} \quad \frac{1}{4} < x \leq \frac{1}{3} \quad x \leq \frac{1}{4}
\]
Weighting Argument for Harmonic Algorithm

Harmonic Algorithm

- Harmonic Algorithm classes: \((\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots, (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}]\).
- Place members of each class separately from others.

\[\text{Harmonic} \quad K = 4\]

\[< 0.9 \ 0.3 \ 0.8 \ 0.5 \ 0.1 \ 0.1 \ 0.3 \ 0.2 \ 0.4 \ 0.2 \ 0.4 \ 0.5 \ 0.5 \ 0.8 \ 0.6 \ 0.4 \ 0.5 \ldots >\]

- \(x > \frac{1}{2}\)
- \(\frac{1}{3} < x \leq \frac{1}{2}\)
- \(\frac{1}{4} < x \leq \frac{1}{3}\)
- \(x \leq \frac{1}{4}\)
Harmonic Algorithm classes: \((\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots, (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}]\).

Place members of each class separately from others.

\[\text{Harmonic} \quad K = 4\]

\[
< 0.9 \ 0.3 \ 0.8 \ 0.5 \ 0.1 \ 0.1 \ 0.3 \ 0.2 \ 0.4 \ 0.2 \ 0.4 \ 0.5 \ 0.5 \ 0.8 \ 0.6 \ 0.4 \ 0.5 \ldots >
\]
Harmonic Algorithm classes: \((\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots, (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}]\).

Place members of each class separately from others.

Harmonic

\[ K = 4 \]

\[
< 0.9 \, 0.3 \, 0.8 \, 0.5 \, 0.1 \, 0.1 \, 0.3 \, 0.2 \, 0.4 \, 0.2 \, 0.4 \, 0.5 \, 0.5 \, 0.8 \, 0.6 \, 0.4 \, 0.5 \ldots >
\]
Harmonic Algorithm classes: \( (\frac{1}{2}, 1], \left(\frac{1}{3}, \frac{1}{2}\right], \ldots, \left(\frac{1}{K}, \frac{1}{K-1}\right], (0, \frac{1}{K}] \).

Place members of each class separately from others.

Harmonic Algorithm: \( K = 4 \)

< 0.9 0.3 0.8 0.5 0.1 0.1 0.3 0.2 0.4 0.2 0.4 0.5 0.5 0.8 0.6 0.4 0.5 ... >
Harmonic Algorithm classes: \((\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots, (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}]\). Place members of each class separately from others.

Harmonic Algorithm

\[\begin{align*}
\text{Harmonic} & \quad K = 4 \\
\downarrow \\
< 0.9 & \quad 0.3 & \quad 0.8 & \quad 0.5 & \quad 0.1 & \quad 0.1 & \quad 0.3 & \quad 0.2 & \quad 0.4 & \quad 0.2 & \quad 0.4 & \quad 0.5 & \quad 0.5 & \quad 0.8 & \quad 0.6 & \quad 0.4 & \quad 0.5 & \ldots >
\end{align*}\]
Harmonic Algorithm

Harmonic Algorithm classes: \((\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots, (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}]\).

Place members of each class separately from others.

Harmonic \(K = 4\)

\(< 0.9 0.3 0.8 0.5 0.1 0.1 0.3 0.2 0.4 0.2 0.4 0.5 0.5 0.8 0.6 0.4 0.5 ... >\)
Harmonic Algorithm classes: \((\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots, (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}].\)

Place members of each class separately from others.

Harmonic \(K = 4\)

\(< 0.9 \ 0.3 \ 0.8 \ 0.5 \ 0.1 \ 0.1 \ 0.3 \ 0.2 \ 0.4 \ 0.2 \ 0.4 \ 0.5 \ 0.5 \ 0.8 \ 0.6 \ 0.4 \ 0.5 \ldots >\)

- \(x > \frac{1}{2}\)
- \(\frac{1}{3} < x \leq \frac{1}{2}\)
- \(\frac{1}{4} < x \leq \frac{1}{3}\)
- \(x \leq \frac{1}{4}\)
Harmonic Algorithm classes: $\left( \frac{1}{2}, 1 \right], \left( \frac{1}{3}, \frac{1}{2} \right], \ldots, \left( \frac{1}{K}, \frac{1}{K-1} \right], \left(0, \frac{1}{K} \right]$. Place members of each class separately from others.

Harmonic $K = 4$

\[ < 0.9 \ 0.3 \ 0.8 \ 0.5 \ 0.1 \ 0.1 \ 0.3 \ 0.2 \ 0.4 \ 0.2 \ 0.4 \ 0.5 \ 0.5 \ 0.8 \ 0.6 \ 0.4 \ 0.5 \ ... > \]
Harmonic Algorithm classes: \( (\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots, (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}] \).

- Place members of each class separately from others.

Harmonic Algorithm

\[ K = 4 \]

\(< 0.9 \ 0.3 \ 0.8 \ 0.5 \ 0.1 \ 0.1 \ 0.3 \ 0.2 \ 0.4 \ 0.2 \ 0.4 \ 0.5 \ 0.5 \ 0.8 \ 0.6 \ 0.4 \ 0.5 \ldots >\]
Harmonic Algorithm classes: \((\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots, (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}]\).

Place members of each class separately from others.

```
Harmonic  K = 4

< 0.9 0.3 0.8 0.5 0.1 0.1 0.3 0.2 0.4 0.2 0.4 0.5 0.5 0.8 0.6 0.4 0.5 ... >
```

```
x > \frac{1}{2}

\frac{1}{3} < x \leq \frac{1}{2}

\frac{1}{4} < x \leq \frac{1}{3}

x \leq \frac{1}{4}
```
Harmonic Algorithm

Harmonic Algorithm classes: \((\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots, (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}]\). Place members of each class separately from others.

Harmonic \(K = 4\)

< 0.9 0.3 0.8 0.5 0.1 0.1 0.3 0.2 0.4 0.2 0.4 0.5 0.5 0.8 0.6 0.4 0.5 ... >
Harmonic Algorithm

Harmonic Algorithm classes: \((\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots, (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}]\).

Place members of each class separately from others.

Harmonic  \(K = 4\)

\(< 0.9 \ 0.3 \ 0.8 \ 0.5 \ 0.1 \ 0.1 \ 0.3 \ 0.2 \ 0.4 \ 0.2 \ 0.4 \ 0.5 \ 0.5 \ 0.8 \ 0.6 \ 0.4 \ 0.5 \ldots >\)
Weighting Argument for Harmonic Algorithm

Harmonic Algorithm

- Harmonic Algorithm classes: \( (\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots, (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}] \).
- Place members of each class separately from others.

Harmonic \( K = 4 \)

\[
< 0.9 \ 0.3 \ 0.8 \ 0.5 \ 0.1 \ 0.1 \ 0.3 \ 0.2 \ 0.4 \ 0.2 \ 0.4 \ 0.5 \ 0.5 \ 0.8 \ 0.6 \ 0.4 \ 0.5 \ldots >
\]

![Diagram](attachment:image.png)
Harmonic Algorithm classes: \( \left( \frac{1}{2}, 1 \right], \left( \frac{1}{3}, \frac{1}{2} \right], \ldots, \left( \frac{1}{K}, \frac{1}{K-1} \right], (0, \frac{1}{K}] \).  
Place members of each class separately from others.

Harmonic Algorithm

\( K = 4 \)

\(< 0.9 \ 0.3 \ 0.8 \ 0.5 \ 0.1 \ 0.1 \ 0.3 \ 0.2 \ 0.4 \ 0.2 \ 0.4 \ 0.5 \ 0.5 \ 0.8 \ 0.6 \ 0.4 \ 0.5 \ ... >\)
Harmonic Algorithm classes: \((\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots, (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}]\).

- Place members of each class separately from others.

Harmonic  \(K = 4\)

\(< 0.9 0.3 0.8 0.5 0.1 0.1 0.3 0.2 0.4 0.2 0.4 0.5 0.5 0.8 0.6 0.4 0.5 ... >\)
Harmonic Algorithm classes: \((\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots, (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}]\).

Place members of each class separately from others.

Harmonic Algorithm

\[ K = 4 \]

\\[
< 0.9 \ 0.3 \ 0.8 \ 0.5 \ 0.1 \ 0.1 \ 0.3 \ 0.2 \ 0.4 \ 0.2 \ 0.4 \ 0.5 \ 0.5 \ 0.8 \ 0.6 \ 0.4 \ 0.5 \ \ldots >
\\
\]
Weighting Technique in a Nutshell

- Assume we want to prove an algorithm Alg is competitive.
- Define a weight $w(x)$ for each item $x$ based on its size.
  - General rule: for an item of size $x$, we should have $w(x) \geq x$. 

Weight should be defined so that total weight of items in any bin (denoted by $w(B)$) is at least 1.

By 'any bin' we mean all bins except possibly a constant number.

Assume algorithm opens $k$ bins; we have $k \cdot 1 \leq W$ where $W$ is the total weight of items in the sequence.

So, we have $\text{Cost}(\text{Alg}) \leq W$ (ignoring a constant no. of bins).

Find the maximum weight of items that fit in any bin.
Let $J$ denote that number.

Opt has to place items with total weight of $W$ into bins each taking weight $J$ out of it.

So, we have $\text{Cost}(\text{Opt}) \geq W / J$.

The competitive ratio of the algorithm will be at most $J$. 

COMP 7720 - Online Algorithms  
Online Bin Packing
Weighting Argument for Harmonic Algorithm

Weighting Technique in a Nutshell

- Assume we want to prove an algorithm Alg is competitive
- Define a weight \( w(x) \) for each item \( x \) based on its size
  - General rule: for an item of size \( x \), we should have \( w(x) \geq x \)
- Weight should be defined so that total weight of items in any bin \( B \) of the algorithm (denoted by \( w(B) \)) is at least 1
  - By ‘any bin’ we mean all bins except possibly a constant number.

\[ \text{Cost}(\text{Alg}) \leq W \text{ (ignoring a constant no. of bins)} \]

\[ \text{Cost}(\text{Opt}) \geq \frac{W}{J} \]

The competitive ratio of the algorithm will be at most \( \frac{J}{6} \).
Weighting Argument for Harmonic Algorithm

Weighting Technique in a Nutshell

- Assume we want to prove an algorithm Alg is competitive
- Define a weight $w(x)$ for each item $x$ based on its size
  - General rule: for an item of size $x$, we should have $w(x) \geq x$
- Weight should be defined so that total weight of items in any bin $B$ of the algorithm (denoted by $w(B)$) is at least 1
  - By ‘any bin’ we mean all bins except possibly a constant number.
  - Assume algorithm opens $k$ bins; we have $k \cdot 1 \leq W$ where $W$ is the total weight of items in the sequence
  - So, we have $\text{Cost}(\text{Alg}) \leq W$ (ignoring a constant no. of bins)
Weighting Argument for Harmonic Algorithm

Weighting Technique in a Nutshell

- Assume we want to prove an algorithm Alg is competitive
- Define a weight \( w(x) \) for each item \( x \) based on its size
  - General rule: for an item of size \( x \), we should have \( w(x) \geq x \)
- Weight should be defined so that total weight of items in any bin \( B \) of the algorithm (denoted by \( w(B) \)) is at least 1
  - By ‘any bin’ we mean all bins except possibly a constant number.
  - Assume algorithm opens \( k \) bins; we have \( k \cdot 1 \leq W \) where \( W \) is the total weight of items in the sequence
  - So, we have \( \text{Cost}(Alg) \leq W \) (ignoring a constant no. of bins)
- Find the maximum weight of items that fit in any bin
  - Let \( J \) denote that number
Weighting Argument for Harmonic Algorithm

Weighting Technique in a Nutshell

- Assume we want to prove an algorithm Alg is competitive
- Define a weight \( w(x) \) for each item \( x \) based on its size
  - General rule: for an item of size \( x \), we should have \( w(x) \geq x \)
- Weight should be defined so that total weight of items in any bin \( B \) of the algorithm (denoted by \( w(B) \)) is at least 1
  - By ‘any bin’ we mean all bins except possibly a constant number.
  - Assume algorithm opens \( k \) bins; we have \( k \cdot 1 \leq W \) where \( W \) is the total weight of items in the sequence
  - So, we have \( \text{Cost}(\text{Alg}) \leq W \) (ignoring a constant no. of bins)
- Find the maximum weight of items that fit in any bin
  - Let \( J \) denote that number
  - \( \text{OPT} \) has to place items with total weight of \( W \) into bins each taking weight \( J \) out of it
  - So, we have \( \text{Cost}(\text{Opt}) \geq W/J \)
Weighting Technique in a Nutshell

- Assume we want to prove an algorithm Alg is competitive.
- Define a weight $w(x)$ for each item $x$ based on its size.
  - General rule: for an item of size $x$, we should have $w(x) \geq x$.
- Weight should be defined so that total weight of items in any bin $B$ of the algorithm (denoted by $w(B)$) is at least 1.
  - By ‘any bin’ we mean all bins except possibly a constant number.
  - Assume algorithm opens $k$ bins; we have $k \cdot 1 \leq W$ where $W$ is the total weight of items in the sequence.
  - So, we have $\text{Cost}(\text{Alg}) \leq W$ (ignoring a constant no. of bins).
- Find the maximum weight of items that fit in any bin.
  - Let $J$ denote that number.
  - $\text{OPT}$ has to place items with total weight of $W$ into bins each taking weight $J$ out of it.
  - So, we have $\text{Cost}(\text{Opt}) \geq W/J$.
- The competitive ratio of the algorithm will be at most $J$. 
Weighting Argument for Harmonic Algorithm

Weighting Technique in a Nutshell

- Step I: Define a weight function \( w(x) \) for item sizes
- Step II: Prove that any bin of the online algorithm has weight 1.
- Step III: Prove that it is not possible to place a total weight more than \( J \) in any empty bin
- The competitive ratio will be \( J \)
Weighting Argument for Harmonic Algorithm

Weighting Technique

- Define a weight for each item based on its size
- The weight of an item in class \( i \) is \( \frac{1}{i} \) when \( i < k \)
- The weight of an item of size \( x \) in class \( k \) is \( \frac{k}{k-1} x \)

Harmonic \( K = 4 \)

\(< 0.9 \ 0.3 \ 0.8 \ 0.5 \ 0.1 \ 0.3 \ 0.2 \ 0.4 \ 0.2 \ 0.4 \ 0.5 \ 0.5 \ 0.8 \ 0.6 \ 0.4 \ 0.5 \ ... > \)

\( x > \frac{1}{2} \)
\( \frac{1}{3} < x \leq \frac{1}{2} \)
\( \frac{1}{4} < x \leq \frac{1}{3} \)
\( x \leq \frac{1}{4} \)

weight = 1 \quad \text{weight} = \frac{1}{2} \quad \text{weight} = \frac{1}{3} \quad \text{weight} = \frac{4}{3} x \)
Weighting Technique for Harmonic

- Total weight of items in each bin of Harmonic is at least 1
- Except possibly the current open bin of each class $\rightarrow k$ bins

$$x > \frac{1}{2}$$
weight = 1

$$\frac{1}{3} < x \leq \frac{1}{2}$$
weight = $\frac{1}{2}$

$$\frac{1}{4} < x \leq \frac{1}{3}$$
weight = $\frac{1}{3}$

$$x \leq \frac{1}{4}$$
weight = $\frac{4}{3}x$
Weighting Technique for Harmonic

- Total weight of items in each bin of Harmonic is at least 1
  - Except possibly the current open bin of each class → $k$ bins
  - Bins of type $i < k$ include $i$ items, each of weight $\frac{1}{i}$ → total weight $i \cdot \frac{1}{i} = 1$
**Weighting Argument for Harmonic Algorithm**

**Weighting Technique for Harmonic**

- Total weight of items in each bin of Harmonic is at least 1
  - Except possibly the current open bin of each class $\rightarrow k$ bins
  - Bins of type $i < k$ include $i$ items, each of weight $\frac{1}{i}$ $\rightarrow$ total weight $i \cdot \frac{1}{i} = 1$
  - Any bin $B$ of type $k$ (except the open bin) has level $> \frac{k-1}{k}$
    - let $y$ be the first item in the next bin opened $\rightarrow y$ did not fit in the previous bin $\rightarrow$ level of the bin + size of $y > 1$ $\Rightarrow$ level of $B > \frac{k-1}{k}$.
Weighting Argument for Harmonic Algorithm

Weighting Technique for Harmonic

- Total weight of items in each bin of Harmonic is at least 1
  - Except possibly the current open bin of each class → \( k \) bins
  - Bins of type \( i < k \) include \( i \) items, each of weight \( \frac{1}{i} \) → total weight \( i \cdot \frac{1}{i} = 1 \)
  - Any bin \( B \) of type \( k \) (except the open bin) has level \( > \frac{k-1}{k} \)
    - let \( y \) be the first item in the next bin opened → \( y \) did not fit in the previous bin → level of the bin + size of \( y > 1 \) \( \frac{y}{1/k} \) level of \( B > \frac{k-1}{k} \).

\[ \text{(Level of } B) > \frac{k-1}{k} \cdot \text{weight of } x = (k-1)/k \cdot x \cdot \text{(total weight of items in } B) > 1 \]

\[ \begin{align*}
  x &> \frac{1}{2} \\
  \text{weight} & = 1 \\
  \begin{array}{cccc}
    0.9 & 0.8 & 0.8 & 0.6 \\
  \end{array}
\end{align*} \]

\[ \begin{align*}
  \frac{1}{3} &< x \leq \frac{1}{2} \\
  \text{weight} & = \frac{1}{2} \\
  \begin{array}{cccc}
    0.4 & 0.5 & 0.4 & \rlap{0.5} \\
  \end{array}
\end{align*} \]

\[ \begin{align*}
  \frac{1}{4} &< x \leq \frac{1}{3} \\
  \text{weight} & = \frac{1}{3} \\
  \begin{array}{cccc}
    0.3 & 0.4 & 0.5 & 0.5 \\
  \end{array}
\end{align*} \]

\[ \begin{align*}
  x &\leq \frac{1}{4} \\
  \text{weight} & = \frac{4}{3} x \\
  \begin{array}{cccc}
    0.2 & 0.2 & 0.1 & 0.1 \\
  \end{array}
\end{align*} \]
Weighting Argument for Harmonic Algorithm

Weighting Technique for Harmonic

How much is the maximum total weight of items in a bin of Opt?

- $x > \frac{1}{2}$: weight = 1
- $\frac{1}{3} < x \leq \frac{1}{2}$: weight = $\frac{1}{2}$
- $\frac{1}{4} < x \leq \frac{1}{3}$: weight = $\frac{1}{3}$
- $x \leq \frac{1}{4}$: weight = $\frac{4}{3}x$

$\rho \leq 2$ for $x > \frac{1}{2}$
$\rho \leq 3/2$ for $\frac{1}{3} < x \leq \frac{1}{2}$
$\rho \leq 4/3$ for $\frac{1}{4} < x \leq \frac{1}{3}$
$\rho = (k + 1)/k = 4/3$ for $x \leq \frac{1}{4}$
How much is the maximum total weight of items in a bin of Opt?

Define density of an item of size $x$ as $\frac{w(x)}{x}$.
Weighting Argument for Harmonic Algorithm

Weighting Technique for Harmonic

How much is the maximum total weight of items in a bin of Opt?

- Define density of an item of size $x$ as $\frac{w(x)}{x}$
- Fill the bin with smallest items of classes.

<table>
<thead>
<tr>
<th>$x &gt; \frac{1}{2}$</th>
<th>$\frac{1}{3} &lt; x \leq \frac{1}{2}$</th>
<th>$\frac{1}{4} &lt; x \leq \frac{1}{3}$</th>
<th>$x \leq \frac{1}{4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>weight = 1</td>
<td>weight = $\frac{1}{2}$</td>
<td>weight = $\frac{1}{3}$</td>
<td>weight = $\frac{4}{3}x$</td>
</tr>
<tr>
<td>$\rho \leq 2$</td>
<td>$\rho \leq \frac{3}{2}$</td>
<td>$\rho \leq \frac{4}{3}$</td>
<td>$\rho = \frac{k+1}{k} = \frac{4}{3}$</td>
</tr>
</tbody>
</table>

COMP 7720 - Online Algorithms  Online Bin Packing 10 / 16
Weighting Argument for Harmonic Algorithm

Weighting Technique for Harmonic

How much is the maximum total weight of items in a bin of Opt?

- Define density of an item of size $x$ as $\frac{w(x)}{x}$
- Fill the bin with smallest items of classes.
- Use a greedy algorithm that places items with a preference for items of higher density (i.e., larger)!

$$\begin{align*}
\text{weight} &= 1 & \rho &\leq 2 \\
\text{weight} &= \frac{1}{2} & \rho &\leq \frac{3}{2} \\
\text{weight} &= \frac{1}{3} & \rho &\leq \frac{4}{3} \\
\text{weight} &= \frac{4}{3}x & \rho &= (k + 1)/k = \frac{4}{3}
\end{align*}$$
How much is the maximum total weight of items in a bin of Opt?

Next largest item that fits: $1/2 + \epsilon$; weight = 1; size = $1/2 + \epsilon$

- Weight = 1, $\rho \leq 2$
- Weight = $1/2$, $\rho \leq 3/2$
- Weight = $1/3$, $\rho \leq 4/3$
- Weight = $4/3 \times x$, $\rho = (k + 1)/k = 4/3$
Weighting Argument for Harmonic Algorithm

Weighting Technique for Harmonic

- How much is the maximum total weight of items in a bin of Opt?
  - Next largest item that fits: $1/2 + \epsilon$; weight = 1; size = $1/2 + \epsilon$
  - Next item that fits: $1/3 + \epsilon$; weight = $1 + \frac{1}{2}$; size = $1/2 + 1/3 + 2\epsilon = \frac{5}{6} + 2\epsilon$

\[
\begin{array}{cccc}
\text{weight} &= 1 & \text{weight} &= \frac{1}{2} \\
\rho &\leq 2 & \rho &\leq 3/2
\end{array}
\]

\[
\begin{array}{cccc}
\text{weight} &= \frac{1}{3} & \text{weight} &= \frac{4}{3}x \\
\rho &\leq 4/3 & \rho &= (k + 1)/k = 4/3
\end{array}
\]
Weighting Argument for Harmonic Algorithm

Weighting Technique for Harmonic

How much is the maximum total weight of items in a bin of Opt?

- Next largest item that fits: $1/2 + \epsilon$; weight = 1; size = $1/2 + \epsilon$
- Next item that fits: $1/3 + \epsilon$; weight = $1 + 1/2$; size = $1/2 + 1/3 + 2\epsilon = 5/6 + 2\epsilon$
- Next item that fits: $1/7 + \epsilon$; weight = $1 + 1/2 + 1/6$; size = $5/6 + 1/7 + 3\epsilon = 41/42 + 3\epsilon$

![Diagram showing weight distribution]

- $x > 1/2$
  - weight = 1
  - $\rho \leq 2$
- $1/3 < x \leq 1/2$
  - weight = $1/2$
  - $\rho \leq 3/2$
- $1/4 < x \leq 1/3$
  - weight = $1/3$
  - $\rho \leq 4/3$
- $x \leq 1/4$
  - weight = $4/3x$
  - $\rho = (k + 1)/k = 4/3$

COMP 7720 - Online Algorithms  Online Bin Packing
Weighting Argument for Harmonic Algorithm

Weighting Technique for Harmonic

How much is the maximum total weight of items in a bin of Opt?

- Next largest item that fits: $1/2 + \epsilon$; weight = 1; size = $1/2 + \epsilon$
- Next item that fits: $1/3 + \epsilon$; weight = $1 + \frac{1}{2}$; size = $1/2 + 1/3 + 2\epsilon = \frac{5}{6} + 2\epsilon$
- Next item that fits: $1/7 + \epsilon$; weight = $1 + \frac{1}{2} + \frac{1}{6}$; size = $\frac{5}{6} + \frac{1}{7} + 3\epsilon = \frac{41}{42} + 3\epsilon$
- Next item that fits: $\frac{1}{43} + \epsilon$; weight = $1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{42}$; size = $\frac{41}{42} + \frac{1}{43} + 4\epsilon$
Weighting Technique for Harmonic

How much is the maximum total weight of items in a bin of Opt?

- So, the greedy approach fills a bin with total weight
  \[ 1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{42} + \frac{1}{(42 \cdot 43)} \ldots \approx 1.691 \]
Weighting Argument for Harmonic Algorithm

Weighting Technique for Harmonic

How much is the maximum total weight of items in a bin of Opt?

- So, the greedy approach fills a bin with total weight
  \[ 1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{42} + \frac{1}{(42 \cdot 43)} \ldots \approx 1.691 \]
- It turns out that it is not possible to achieve higher weight
Weighting Argument for Harmonic Algorithm

Weighting Technique for Harmonic

How much is the maximum total weight of items in a bin of Opt?

So, the greedy approach fills a bin with total weight:

\[ 1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{42} + \frac{1}{(42 \cdot 43)} \ldots \approx 1.691 \]

It turns out that it is not possible to achieve higher weight:

E.g., if there is no item of class 1, the density and hence total weight will be less than \( \frac{3}{2} \) → there is an item of size \( \frac{1}{2} + \epsilon \).
Weighting Argument for Harmonic Algorithm

Weighting Technique for Harmonic

How much is the maximum total weight of items in a bin of Opt?

- So, the greedy approach fills a bin with total weight
  \[ 1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{42} + \frac{1}{(42 \cdot 43)} \ldots \approx 1.691 \]
- It turns out that it is not possible to achieve higher weight
  - E.g., if there is no item of class 1, the density and hence total weight will be less than \( \frac{3}{2} \implies \) there is an item of size \( \frac{1}{2} + \epsilon \)
  - If there is an item of size \( \frac{1}{2} + \epsilon \) and no item of class 2, there can be at most one item \( \frac{1}{4} + \epsilon \) of class 3, and density of the rest is less than \( \frac{5}{4} \). Weight will be \( 1 + \frac{1}{3} + \frac{5}{4} \cdot \frac{1}{4} \approx 1.64 \implies \) there is an item of size \( \frac{1}{3} + \epsilon \)
Weighting Argument for Harmonic Algorithm

Summary of Weighting Technique for Harmonic

- We define a weight of an item of class $i < k$ to be $1/i$ and the weight of an item of class $k$ to be $k / (k-1) \cdot x$.

- We showed the weight of all bins (except at most $k$ of them) is at least 1 in Harmonic’s packing.

- We showed the maximum weight of any bin is at most $J = 1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{42} + \ldots$ when $k$ is large enough.
  - We often assume $k$ is a constant around 20.

- The competitive ratio of the algorithm will be at most $J$. 
Consider the following sequence

\[
\langle 1/43 + \epsilon, \ldots, 1/43 + \epsilon, 1/7 + \epsilon, \ldots, 1/7 + \epsilon, 1/3 + \epsilon, \ldots, 1/3 + \epsilon, 1/2 + \epsilon, \ldots, 1/2 + \epsilon, \rangle
\]

Harmonic opens \( m(1/42 + 1/6 + 1/2 + 1) \approx 1.691m \) bins

Opt places one item of each class in each bin → \( m \) bins
Consider the following sequence

\[
\langle \frac{1}{43} + \epsilon, \ldots, \frac{1}{43} + \epsilon, \frac{1}{7} + \epsilon, \ldots, \frac{1}{7} + \epsilon, \frac{1}{3} + \epsilon, \ldots, \frac{1}{3} + \epsilon, \frac{1}{2} + \epsilon, \ldots, \frac{1}{2} + \epsilon, \rangle
\]

- Harmonic opens \( m(1/42 + 1/6 + 1/2 + 1) \approx 1.691m \) bins
- Opt places one item of each class in each bin \( \rightarrow m \) bins
Weighting Argument for Harmonic Algorithm

Lower Bound: a Nasty Sequence

Consider the following sequence

\[
\langle 1/43 + \epsilon, \ldots, 1/43 + \epsilon, 1/7 + \epsilon, \ldots, 1/7 + \epsilon, 1/3 + \epsilon, \ldots, 1/3 + \epsilon, 1/2 + \epsilon, \ldots, 1/2 + \epsilon, \rangle
\]

What about First Fit and Best Fit?

Both create the same packing as Harmonic!
Weighting Argument for Harmonic Algorithm

Lower Bound: a Nasty Sequence

Consider the following sequence

\[\langle 1/43 + \epsilon, \ldots, 1/43 + \epsilon, 1/7 + \epsilon, \ldots, 1/7 + \epsilon, 1/3 + \epsilon, \ldots, 1/3 + \epsilon, 1/2 + \epsilon, \ldots, 1/2 + \epsilon, \rangle\]

- What about First Fit and Best Fit?
- Both create the same packing as Harmonic!
Weighting Argument for Harmonic Algorithm

Summary

- Competitive ratio Harmonic is $j = 1.691$.
- Competitive ratios of Best Fit and First Fit is at least $J$
- Indeed their ratio is 1.7
- We see a sketch of the proof in the next class