COMP 7720 - Online Algorithms

Online Bin Packing

Shahin Kamali

Lecture 21 - Nov. 22nd, 2018

University of Manitoba
Today’s plan

- Online bin packing with advice (finishing bin packing)
- Advice on giving a research talk (if we have time)
Review & Plan

Breaking the Lower Bound

- Receive number of critical items with $O(\log n)$ bits of advice
- Consider **ReserveCritical**:
  - At the beginning, reserve a space of size $2/3$ for critical items
  - **huge** items: open a new bin
  - **critical** items: place in a reserve space
  - **mini** item: place two of them in the same bin
  - **tiny** items: apply First-Fit to place in bins with critical or other tiny items
ReserveCritical Algorithm

At the beginning, reserve a space of size $2/3$ for critical items

- **Huge** items: open a new bin (no other item goes there)
- **critical** items: place in a reserve space
- **mini** item: place two of them in the same bin (no other item goes there)
- **tiny** items: apply FF to place in bins with critical or other tiny items

$$\sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \rangle$$
At the beginning, reserve a space of size 2/3 for critical items

- **Huge** items: open a new bin (no other item goes there)
- **critical** items: place in a reserve space
- **mini** item: place two of them in the same bin (no other item goes there)
- **tiny** items: apply FF to place in bins with critical or other tiny items

\[ \sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \ \rangle \]
ReserveCritical Algorithm

At the beginning, reserve a space of size 2/3 for critical items

- **Huge** items: open a new bin (no other item goes there)
- **critical** items: place in a reserve space
- **mini** item: place two of them in the same bin (no other item goes there)
- **tiny** items: apply FF to place in bins with critical or other tiny items

\[ \sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \ \rangle \]
At the beginning, reserve a space of size $2/3$ for critical items

- **Huge** items: open a new bin (no other item goes there)
- **critical** items: place in a reserve space
- **mini** item: place two of them in the same bin (no other item goes there)
- **tiny** items: apply FF to place in bins with critical or other tiny items

$$\sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \rangle$$
ReserveCritical Algorithm

At the beginning, reserve a space of size 2/3 for critical items

- **Huge** items: open a new bin (no other item goes there)
- **critical** items: place in a reserve space
- **mini** item: place two of them in the same bin (no other item goes there)
- **tiny** items: apply FF to place in bins with critical or other tiny items

$$\sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \rangle$$
Review & Plan

ReserveCritical Algorithm

At the beginning, reserve a space of size 2/3 for critical items

- **Huge** items: open a new bin (no other item goes there)
- **critical** items: place in a reserve space
- **mini** item: place two of them in the same bin (no other item goes there)
- **tiny** items: apply FF to place in bins with critical or other tiny items

\[ \sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \ \rangle \]
At the beginning, reserve a space of size 2/3 for critical items

- **Huge** items: open a new bin (no other item goes there)
- **critical** items: place in a reserve space
- **mini** item: place two of them in the same bin (no other item goes there)
- **tiny** items: apply FF to place in bins with critical or other tiny items

\[ \sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \ \rangle \]
ReserveCritical Algorithm

- At the beginning, reserve a space of size 2/3 for critical items
  - **Huge** items: open a new bin (no other item goes there)
  - **critical** items: place in a reserve space
  - **mini** item: place two of them in the same bin (no other item goes there)
  - **tiny** items: apply FF to place in bins with critical or other tiny items

$$\sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \rangle$$
ReserveCritical Algorithm

At the beginning, reserve a space of size 2/3 for critical items

- **Huge** items: open a new bin (no other item goes there)
- **critical** items: place in a reserve space
- **mini** item: place two of them in the same bin (no other item goes there)
- **tiny** items: apply FF to place in bins with critical or other tiny items

\[ \sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \ \rangle \]
At the beginning, reserve a space of size $2/3$ for critical items

- **Huge** items: open a new bin (no other item goes there)
- **critical** items: place in a reserve space
- **mini** item: place two of them in the same bin (no other item goes there)
- **tiny** items: apply FF to place in bins with critical or other tiny items

$$\sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \rangle$$
At the beginning, reserve a space of size 2/3 for critical items

- **Huge** items: open a new bin (no other item goes there)
- **critical** items: place in a reserve space
- **mini** item: place two of them in the same bin (no other item goes there)
- **tiny** items: apply FF to place in bins with critical or other tiny items

\[ \sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \ \rangle \]
Reserve Critical Algorithm

At the beginning, reserve a space of size 2/3 for critical items

- **Huge** items: open a new bin (no other item goes there)
- **critical** items: place in a reserve space
- **mini** item: place two of them in the same bin (no other item goes there)
- **tiny** items: apply FF to place in bins with critical or other tiny items

\[ \sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \rangle \]
Reserve Critical Algorithm

- At the beginning, reserve a space of size 2/3 for critical items
  - **Huge** items: open a new bin (no other item goes there)
  - **critical** items: place in a reserve space
  - **mini** item: place two of them in the same bin (no other item goes there)
  - **tiny** items: apply FF to place in bins with critical or other tiny items

\[
\sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \rangle
\]
 Reserve Critical Algorithm

- At the beginning, reserve a space of size 2/3 for critical items
  - **Huge** items: open a new bin (no other item goes there)
  - **critical** items: place in a reserve space
  - **mini** item: place two of them in the same bin (no other item goes there)
  - **tiny** items: apply FF to place in bins with critical or other tiny items

\[ \sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \ \rangle \]
At the beginning, reserve a space of size 2/3 for critical items

- **Huge** items: open a new bin (no other item goes there)
- **critical** items: place in a reserve space
- **mini** item: place two of them in the same bin (no other item goes there)
- **tiny** items: apply FF to place in bins with critical or other tiny items

\[ \sigma = \langle 0.3, 0.9, 0.6, 0.5, 0.1, 0.1, 0.56, 0.4, 0.3, 0.45, 0.8, 0.51, 0.41, 0.2, 0.1, 0.37, 0.3 \rangle \]
Reserve Critical Algorithm

At the beginning, reserve a space of size $2/3$ for critical items

- **Huge** items: open a new bin (no other item goes there)
- **critical** items: place in a reserve space
- **mini** item: place two of them in the same bin (no other item goes there)
- **tiny** items: apply FF to place in bins with critical or other tiny items

$$\sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \rangle$$
Reserve Critical Algorithm

At the beginning, reserve a space of size 2/3 for critical items

- **Huge** items: open a new bin (no other item goes there)
- **critical** items: place in a reserve space
- **mini** item: place two of them in the same bin (no other item goes there)
- **tiny** items: apply FF to place in bins with critical or other tiny items

\[
\sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \rangle
\]
At the beginning, reserve a space of size 2/3 for critical items

- **Huge** items: open a new bin (no other item goes there)
- **critical** items: place in a reserve space
- **mini** item: place two of them in the same bin (no other item goes there)
- **tiny** items: apply FF to place in bins with critical or other tiny items

\[ \sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \rangle \]
Reserve Critical Algorithm

- At the beginning, reserve a space of size 2/3 for critical items
  - **Huge** items: open a new bin (no other item goes there)
  - **critical** items: place in a reserve space
  - **mini** item: place two of them in the same bin (no other item goes there)
  - **tiny** items: apply FF to place in bins with critical or other tiny items

\[ \sigma = \langle 0.3 \ 0.9 \ 0.6 \ 0.5 \ 0.1 \ 0.1 \ 0.56 \ 0.4 \ 0.3 \ 0.45 \ 0.8 \ 0.51 \ 0.41 \ 0.2 \ 0.1 \ 0.37 \ 0.3 \rangle \]
ReserveCritical Analysis

1. ReserveCritical has c.r. of at most 1.5.

2. Based on the final packing, we considered two cases:
   - Case 1: There is a bin opened by a tiny item → all bins have level at least 2/3
   - Case 2: No bin is opened by a tiny item → using a weighting argument!
Review & Plan

ReserveCritical algorithm

Theorem

*Competitive ratio of ReserveCritical is at most 1.5.*

- With $O(\log n)$ bits of advice, one can achieve a competitive ratio of 1.5
- Can we improve this?
RedBlue Algorithm (sketch)

Instead of receiving the number of critical items in $O(\log n)$ bits, receive the ratio between critical and tiny bins in the final packing of ReserveCritical.
RedBlue Algorithm (sketch)

- Instead of receiving the number of critical items in $O(\log n)$ bits, receive the ratio between critical and tiny bins in the final packing of ReserveCritical.

- Treat Huge and Mini items as before.
RedBlue Algorithm (sketch)

- Instead of receiving the number of critical items in $O(\log n)$ bits, receive the ratio between critical and tiny bins in the final packing of ReserveCritical

- Treat Huge and Mini items as before

- Place a critical item in the reserved space of a critical bin; if no reserve space exists, open a new bin and declare it as critical

- Place a tiny item in non-reserved space of critical bins (using FF)

- If no such critical bin exists, place it in the available space of a tiny bin (using Next Fit)

- If no suitable tiny bin exists, open a new bin

- Declare the new bin to be a critical or a tiny bin so that the ratio between the number of these bins becomes closer to the ratio received in advice
RedBlue Algorithm (sketch)

- Instead of receiving the number of critical items in $O(\log n)$ bits, receive the ratio between critical and tiny bins in the final packing of ReserveCritical.

- Treat Huge and Mini items as before.

- Place a critical item in the reserved space of a critical bin; if no reserve space exists, open a new bin and declare it as critical.

- Place a tiny item in non-reserved space of critical bins (using FF).
  - If no such critical bin exists, place it in the available space of a tiny bin (using Next Fit).
RedBlue Algorithm (sketch)

- Instead of receiving the number of critical items in $O(\log n)$ bits, receive the ratio between critical and tiny bins in the final packing of ReserveCritical.

- Treat Huge and Mini items as before.

- Place a critical item in the reserved space of a critical bin; if no reserve space exists, open a new bin and declare it as critical.

- Place a tiny item in non-reserved space of critical bins (using FF).
  - If no such critical bin exists, place it in the available space of a tiny bin (using Next Fit).
  - If no suitable tiny bin exists, open a new bin.
RedBlue Algorithm (sketch)

Instead of receiving the number of critical items in $O(\log n)$ bits, receive the ratio between critical and tiny bins in the final packing of ReserveCritical.

Treat Huge and Mini items as before.

Place a critical item in the reserved space of a critical bin; if no reserve space exists, open a new bin and declare it as critical.

Place a tiny item in non-reserved space of critical bins (using FF):
- If no such critical bin exists, place it in the available space of a tiny bin (using Next Fit).
- If no suitable tiny bin exists, open a new bin.
- Declare the new bin to be a critical or a tiny bin so that the ratio between the number of these bins becomes closer to the ratio received in advice.
RedBlue Algorithm (sketch)

If the ratio between critical and tiny bins is encoded using $k$ bits of advice, RedBlue algorithm has a competitive ratio of at most
\[1.5 + \frac{15}{2^{k/2+1}}\]

Theorem

With constant number of bits of advice, one can achieve a competitive ratio of (almost) 1.5.
RedBlue Algorithm (sketch)

If the ratio between critical and tiny bins is encoded using $k$ bits of advice, RedBlue algorithm has a competitive ratio of at most $1.5 + \frac{15}{2^{k/2+1}}$.

Theorem

*With constant number of bits of advice, one can achieve a competitive ratio of (almost) 1.5.*

Can we do better?
In fact, with a more complicated argument, we can show that with advice of constant size, one can achieve a competitive ratio of 1.47.
In fact, with a more complicated argument, we can show that with advice of constant size, one can achieve a competitive ratio of 1.47.

Idea: pack items of size larger than 1/3 separately from the rest.

How advice can help in packing items of size larger than 1/3?
It is often useful to think of algorithms that ‘complement’ each other.
Power of Advice of Constant Size

- It is often useful to think of algorithms that ‘complement’ each other.

Assume all items are larger than 1/3:

- **Sbf**: All small items ($< 1/2$) are packed according to BestFit, and each large item ($\geq 1/2$) is placed in a new bin.
- **Lbf**: All large items are packed according to BestFit, and each small item is placed in a new bin.

\[ \sigma = \langle 0.45, 0.6, 0.75, 0.34, 0.40, 0.56, 0.35, 0.55, 0.50 \rangle \]
Power of Advice of Constant Size

- It is often useful to think of algorithms that ‘complement’ each other.
- Assume all items are larger than 1/3:
  - **Sbf**: All small items \(< 1/2\) are packed according to BestFit, and each large item \(\geq 1/2\) is placed in a new bin.
  - **Lbf**: All large items are packed according to BestFit, and each small item is placed in a new bin.

\[ \sigma = \langle 0.45, 0.6, 0.75, 0.34, 0.40, 0.56, 0.35, 0.55, 0.50 \rangle \]
Power of Advice of Constant Size

**Theorem**

*When all items are larger than $1/3$, the better algorithm among Sbf and Lbf has a competitive ratio of 1.39.*

- With only one bit of advice, one can achieve a competitive ratio of 1.39 (when all items are larger than $1/3$).
- Think of the two algorithms as ‘parallel algorithms’
- This algorithm is used as a subroutine for an algorithm which gets a competitive ratio of 1.47 with constant advice! (details skipped here).
Lower bound

- Advice of size $\Omega(n)$ is required to achieve an algorithm with c.r. $\leq 9/8$
Lower bound

- Advice of size $\Omega(n)$ is required to achieve an algorithm with c.r. $\leq 9/8$
- A reduction from binary guessing problem
Review & Plan

Lower bound

- Advice of size $\Omega(n)$ is required to achieve an algorithm with c.r. $\leq 9/8$
- A reduction from binary guessing problem
- Consider

$$\sigma = \langle 0.5 + \epsilon, \ldots 0.5 + \epsilon, \ a_1, a_2, a_3 \ldots, a_{2m}, \ b_1, \ldots b_m \rangle$$

- $m$ green items
- $m$ white items in range $(1/3, 1/2]$ complements of smaller white items
Advice of size $\Omega(n)$ is required to achieve an algorithm with c.r. $\leq 9/8$.

A reduction from binary guessing problem.

Consider

$$\sigma = \langle 0.5 + \epsilon, \ldots 0.5 + \epsilon, \quad a_1, a_2, a_3 \ldots, a_{2m}, \quad b_1, \ldots b_m \rangle$$

$m$ green items, white items in range $(1/3, 1/2]$ complements of smaller white items

The algorithm should ‘guess’ whether each white item is among the larger half of smaller half of white items!
**Lower bound**

\[
\langle 0.51, \ldots, 0.51, 0.42, 0.37, 0.4, 0.39, 0.38, 0.385, 0.388, 0.386, 0.63, 0.62, 0.615, 0.61 \rangle
\]

- \(m\) green items
- \(2m\) white items
- red complements

Guess if an item is among smaller or larger half of white items

Open a new bin for smaller half of white items (in anticipation of their complements coming in the future)

For the larger half of white items, put them with green items

The 'type' (being in smaller or larger half) of the white item cannot be revealed from knowing types of previous white items

For any four mistakes in guessing, at least 1 extra bin is opened.
Review & Plan

Lower bound

\[ \langle 0.51, \ldots, 0.51, 0.42, 0.37, 0.4, 0.39, 0.38, 0.385, 0.388, 0.386, 0.63, 0.62, 0.615, 0.614 \rangle \]

\( m \) green items

\( 2m \) white items

red complements

Guess if an item is among smaller or larger half of white items

open a new bin for smaller half of white items (in anticipation of their complements coming in the future)

for the larger half of white items, put them with green items

The 'type' (being in smaller or larger half) of the white item cannot be revealed from knowing types of previous white items

For any four mistakes in guessing, at least 1 extra bin is opened
**Review & Plan**

**Lower bound**

\[
\langle 0.51, \ldots, 0.51, 0.42, 0.37, 0.4, 0.39, 0.38, 0.385, 0.388, 0.386, \ldots, 0.63, 0.62, 0.615, 0.614 \rangle
\]

- **Guess if an item is among smaller or larger half of white items**
  - open a new bin for smaller half of white items (in anticipation of their complements coming in the future)
  - for the larger half of white items, put them with green items

---

**COMP 7720 - Online Algorithms**

**Online Bin Packing**
Review & Plan

Lower bound

\[
\langle 0.51, \ldots, 0.51, 0.42, 0.37, 0.4, 0.39, 0.38, 0.385, 0.388, 0.386, 0.63, 0.62, 0.615, 0.614 \rangle
\]

- \( m \) green items
- \( 2m \) white items
- red complements

- Guess if an item is among smaller or larger half of white items
  - open a new bin for smaller half of white items (in anticipation of their complements coming in the future)
  - for the larger half of white items, put them with green items
- The ‘type’ (being in smaller or larger half) of the white item cannot be revealed from knowing types of previous white items
- For any four mistakes in guessing, at least 1 extra bin is opened
Theorem

In order to achieve a competitive ratio better than $9/8$, advice of linear size is required

This result can be improved to show that for a competitive ratio better than $4 - 2\sqrt{2} \approx 1.172$, a linear number of bits are required.
Bin Packing & Advice: A General Picture

- No advice: best upper and lower bounds by [Heydrich and van Stee, 2015] and [Balogh et al., 2012].
- With $\Theta(n \log N)$ bits, one can achieve an optimal solution ($N$ is the cost of $\text{OPT}$) [Boyar et al., 2014].

\[ \text{competitive ratio} \]

\[ 1.7 \]

\[ 1.5817 \]

\[ 1.5403 \]

\[ 1.172 \]

\[ 1.666 \]

$\Theta (n \log N)$

advice size
- With $\Theta(\log n)$ bits, one can achieve a competitive ratio of 1.5 (better than all online algorithms) [Boyar et al., 2014].
- With linear number bits, one can achieve a competitive ratio of 4/3 [Boyar et al., 2014].
For a competitive ratio better than $9/8$, a linear number of bits are required [Boyar et al., 2014].
Bin Packing & Advice: A General Picture

- With linear number bits, one can achieve a competitive ratio of 1.0 [Renault et al., 2014].
- With $k \geq 4$ bits, one can get a competitive ratio of $1.5 + \frac{15}{2^{k/2}+1}$ [Angelopoulos et al., 2015].
- With $\Theta(1)$ bits, one can get a competitive ratio of 1.4702 [Angelopoulos et al., 2015].
Bin Packing & Advice: A General Picture

- For a competitive ratio better than $7/6$, a linear number of bits are required [Angelopoulos et al., 2015].
- For a competitive ratio better than $4 - 2\sqrt{2} \approx 1.172$, a linear number of bits are required [Mikkelsen, 2015].
References

"Online Bin Packing with Advice of Small Size".

"New lower bounds for certain classes of bin packing algorithms".

"Online Bin Packing with Advice".
pages 174–186.

"Beating the Harmonic lower bound for online bin packing".
CoRR, abs/1511.00876.

"Optimal Online Edge Coloring of Planar Graphs with Advice".

"Online Algorithms with Advice for Bin Packing and Scheduling Problems".
CoRR, abs/1311.7589.

"Online Bin Packing with Advice of Small Size".


