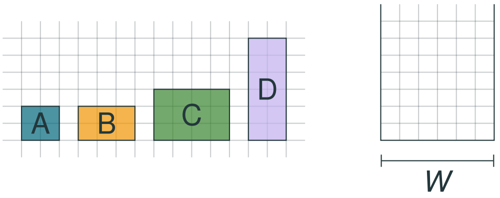
A Tight $(3/2 + \varepsilon)$ Approximation for Skewed Strip Packing

W. Gálvez, F. Grandoni, A. Jabal Ameli, K. Jansen, A. Khan and M. Rau

Introduction

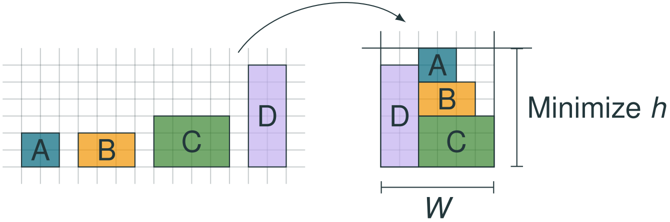
Strip packing

- Input:
  - Strip with width $W \in \mathbb{N}$ and infinite height
  - Set of n items $\mathcal{I}$ with width $w_i \in \mathbb{N}_{\leq W}$ and height $h_i \in \mathbb{N}$



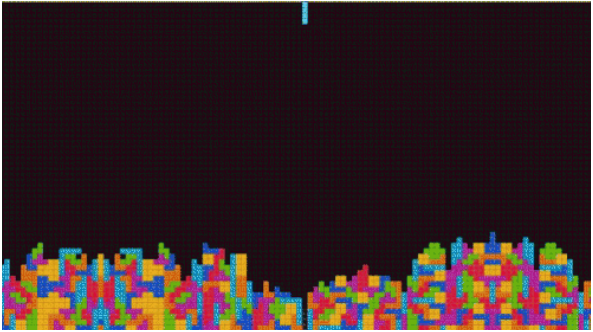
Strip packing

- Input:
  - Strip with width $W \in \mathbb{N}$ and infinite height
  - Set of n items $\mathcal{I}$ with width $w_i \in \mathbb{N}_{\leq W}$ and height $h_i \in \mathbb{N}$
- Objective:
  - Find a *feasible* packing of the items in the strip with minimum height.



Strip Packing

Strip Packing bears a resemblance to Tetris.

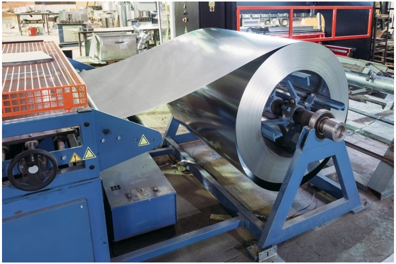


Applications

- Manufacturing Processes

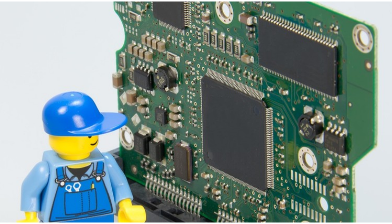
Applications

- Manufacturing Processes



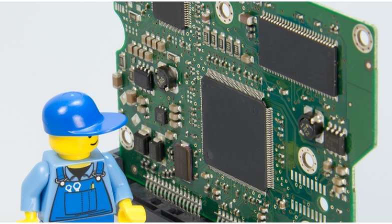
Applications

- VLSI Design



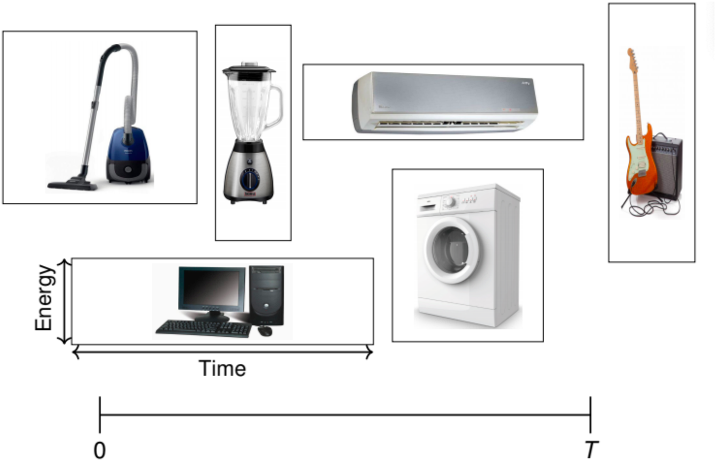
Applications

- VLSI Design



Applications

- Optimizing Energy Consumption



Applications

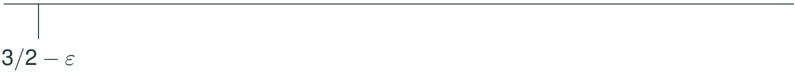
- Optimizing Energy Consumption

State Of the Art

$3/2 - \varepsilon$

There is no $(3/2 - \varepsilon)$-approximation unless $P = NP$

State Of the Art



There is no $(3/2 - \varepsilon)$**-approximation unless** $P = NP$

By a simple reduction from the *Partition* problem.

State Of the Art

There is no $(3/2 - \varepsilon)$-approximation unless $P = NP$

By a simple reduction from the *Partition* problem.

Approximation Algorithms:

[1] B.S. Baker, D. J. Brown and H. P. Katseff, 1980

[2] E.G. Coffman Jr., M. R. Garey, D. S. Johnson and R. E. Tarjan, 1980

[3] D. Sleator, 1980

State Of the Art

There is no $(3/2 - \varepsilon)$-approximation unless $P = NP$

By a simple reduction from the *Partition* problem.

Approximation Algorithms:

[1] B.S. Baker, D. J. Brown and H. P. Katseff, 1980

[2] E.G. Coffman Jr., M. R. Garey, D. S. Johnson and R. E. Tarjan, 1980

[3] D. Sleator, 1980

[4] A. Steinberg, 1997

[5] I. Schiermeyer, 1994

State Of the Art

There is no $(3/2 - \varepsilon)$-approximation unless $P = NP$

By a simple reduction from the *Partition* problem.

Approximation Algorithms:

[1] B.S. Baker, D. J. Brown and H. P. Katseff, 1980

[2] E.G. Coffman Jr., M. R. Garey, D. S. Johnson and R. E. Tarjan, 1980

[3] D. Sleator, 1980

[4] A. Steinberg, 1997

[5] I. Schiermeyer, 1994

[6] R. Harren and R. van Stee, 2009

State Of the Art

There is no $(3/2-\varepsilon)$-approximation unless $P = NP$

By a simple reduction from the *Partition* problem.

Approximation Algorithms:

[1] B.S. Baker, D. J. Brown and H. P. Katseff, 1980

[2] E.G. Coffman Jr., M. R. Garey, D. S. Johnson and R. E. Tarjan, 1980

[3] D. Sleator, 1980

[4] A. Steinberg, 1997

[5] I. Schiermeyer, 1994

[6] R. Harren and R. van Stee, 2009

[7] R. Harren, K. Jansen, L. Prädel and R. van Stee, 2014

Pseudo-Polynomial Time $(n \cdot W \cdot h_{\max})^{\mathcal{O}(1)}$

State Of the Art

Pseudo-Polynomial Time $(n \cdot W \cdot h_{\max})^{\mathcal{O}(1)}$

Algorithms

$3/2 + \varepsilon$

1

[1]

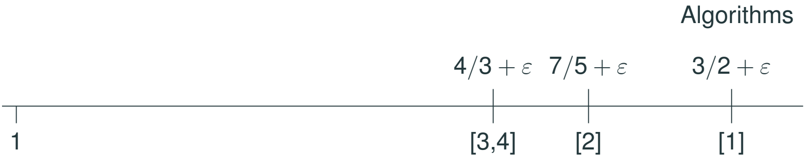
[1] K. Jansen and R. Thöle, 2008

Pseudo-Polynomial Time $(n \cdot W \cdot h_{\max})^{\mathcal{O}(1)}$

[1] K. Jansen and R. Thöle, 2008
[2] G. Nadiradze and A. Wiese, 2016

State Of the Art

Pseudo-Polynomial Time $(n \cdot W \cdot h_{\max})^{\mathcal{O}(1)}$



[1] K. Jansen and R. Thöle, 2008
[2] G. Nadiradze and A. Wiese, 2016
[3] W. Gálvez, F. Grandoni, S. Ingala and A. Khan, 2016
[4] K. Jansen and M. Rau, 2017

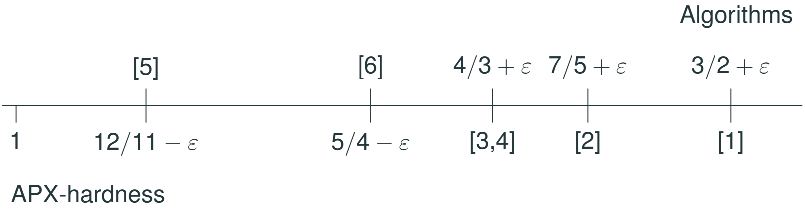
State Of the Art

Pseudo-Polynomial Time $(n \cdot W \cdot h_{\max})^{\mathcal{O}(1)}$

[1] K. Jansen and R. Thöle, 2008
[2] G. Nadiradze and A. Wiese, 2016
[3] W. Gálvez, F. Grandoni, S. Ingala and A. Khan, 2016
[4] K. Jansen and M. Rau, 2017
[5] A. Adamaszek, T. Kociumaka, M. Pilipczuk and M. Pilipczuk, 2017

State Of the Art

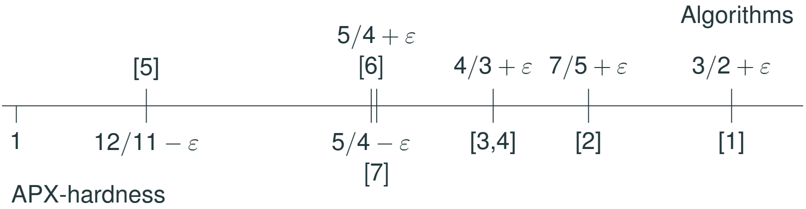
Pseudo-Polynomial Time $(n \cdot W \cdot h_{\max})^{\mathcal{O}(1)}$



[1] K. Jansen and R. Thöle, 2008
[2] G. Nadiradze and A. Wiese, 2016
[3] W. Gálvez, F. Grandoni, S. Ingala and A. Khan, 2016
[4] K. Jansen and M. Rau, 2017
[5] A. Adamaszek, T. Kociumaka, M. Pilipczuk and M. Pilipczuk, 2017
[6] S. Henning, K. Jansen, M. Rau and L. Schmarje, 2018

State Of the Art

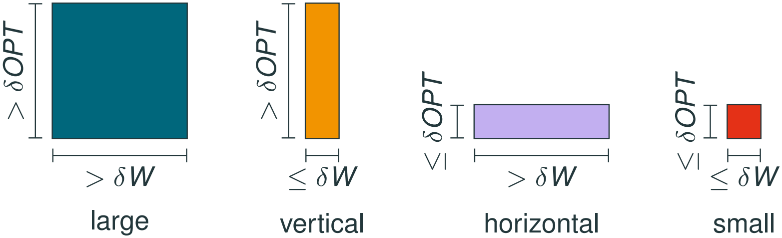
Pseudo-Polynomial Time $(n \cdot W \cdot h_{\max})^{\mathcal{O}(1)}$



[1] K. Jansen and R. Thöle, 2008
[2] G. Nadiradze and A. Wiese, 2016
[3] W. Gálvez, F. Grandoni, S. Ingala and A. Khan, 2016
[4] K. Jansen and M. Rau, 2017
[5] A. Adamaszek, T. Kociumaka, M. Pilipczuk and M. Pilipczuk, 2017
[6] S. Henning, K. Jansen, M. Rau and L. Schmarje, 2018
[7] K. Jansen and M. Rau, 2019

Standard Approach: Size classification

Classify items by size



Large items

Observation

- *There can be only* $\mathcal{O}_\delta(1)$ *large items.*
- *If we are given an instance containing only large items we are able to compute efficiently.*

Large items

Observation

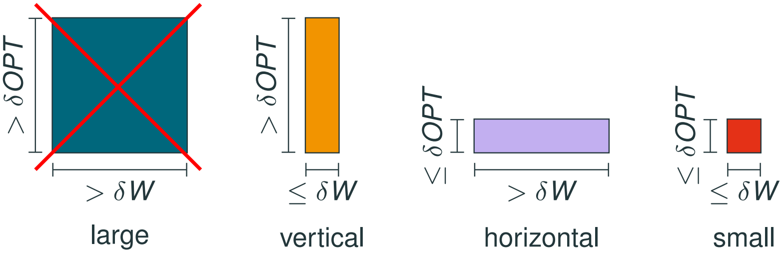
- *There can be only $\mathcal{O}_\delta(1)$ large items.*
- *If we are given an instance containing only large items we are able to compute efficiently.*

How about the complementary case?

Our problem

δ-skewed Strip Packing

Strip Packing restricted to instances where all items have either $w_i \leq \delta W$ or $h_i \leq \delta T$.



Our Results

Theorem

For any $\delta > 0$ and $\varepsilon > 0$, there is no polynomial time $(3/2 - \varepsilon)$-approximation for δ-skewed Strip Packing unless $P = NP$.

Our Results

Theorem

For any $\delta > 0$ and $\varepsilon > 0$, there is no polynomial time $(3/2 - \varepsilon)$-approximation for δ-skewed Strip Packing unless $P = NP$.

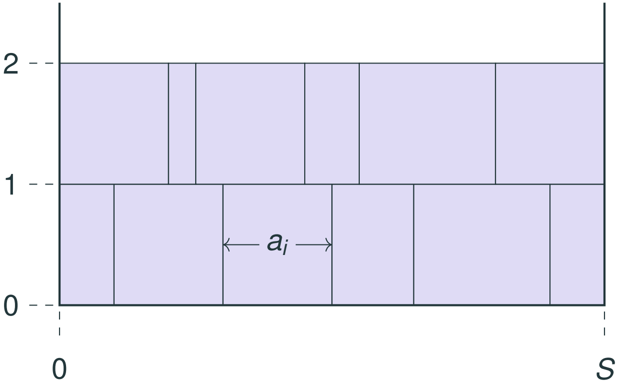
Theorem

For any $\varepsilon > 0$ there exists $\delta > 0$ such that there is a $(3/2 + \varepsilon)$-approximation for δ-skewed Strip Packing.

Hardness

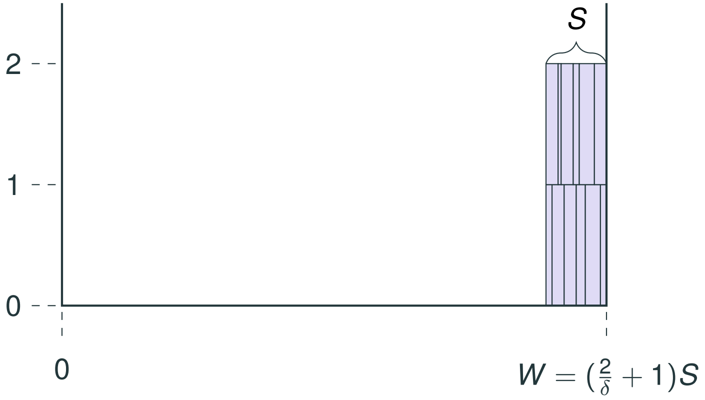
Reduction from Partition

- Classical reduction from Partition: $2n$ rectangles of height 1 and widths $a_1, a_2, \ldots, a_{2n}$ satisfying $a_1 + \cdots + a_{2n} = 2S$.



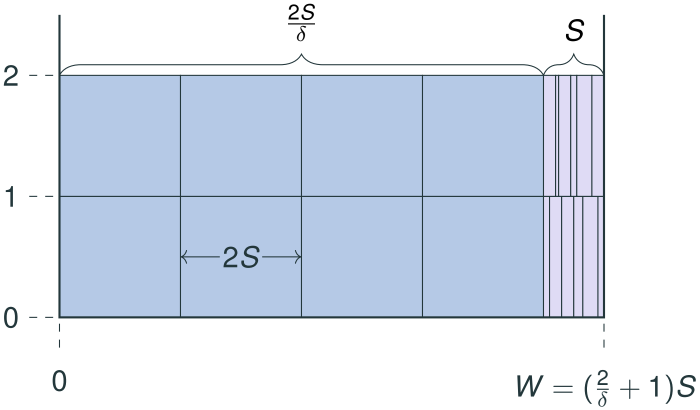
Reduction from Partition

- Let us increase the width of the strip by a factor $2/\delta + 1$; this way the rectangles become δ-skewed.



Reduction from Partition

- Let us add δ-skewed dummy items of width $2S$ and height 1 (they are δ-skewed).



Algorithm

General approach of previous algorithms

Search for well-structured solutions

Decompose the solution into *simple* regions where items are packed *by means of polynomial time algorithms*.

General approach of previous algorithms

Search for well-structured solutions

Decompose the solution into *simple* regions where items are packed *by means of polynomial time algorithms*.

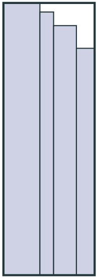
Example: Container packings.

General approach of previous algorithms

Search for well-structured solutions

Decompose the solution into *simple* regions where items are packed *by means of polynomial time algorithms*.

Example: Container packings.

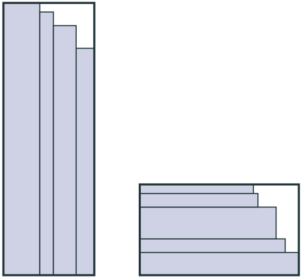


General approach of previous algorithms

Search for well-structured solutions

Decompose the solution into *simple* regions where items are packed *by means of polynomial time algorithms*.

Example: Container packings.

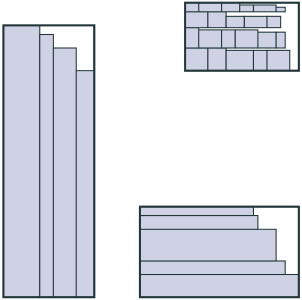


General approach of previous algorithms

Search for well-structured solutions

Decompose the solution into *simple* regions where items are packed *by means of polynomial time algorithms*.

Example: Container packings.

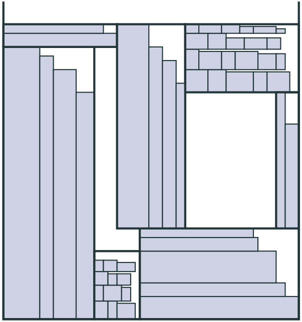


General approach of previous algorithms

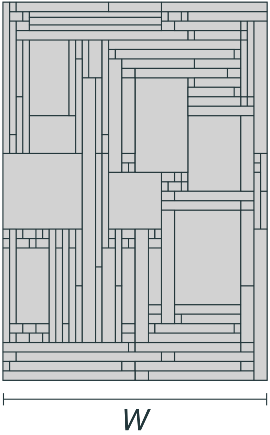
Search for well-structured solutions

Decompose the solution into *simple* regions where items are packed *by means of polynomial time algorithms*.

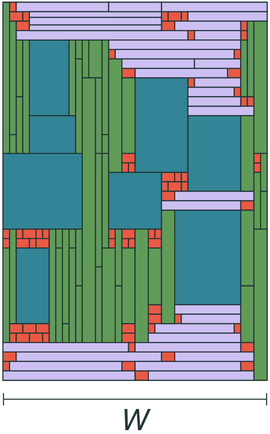
Example: Container packings.



Harren et al.'s algorithm

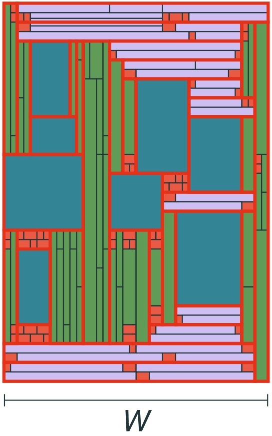


1. Consider the optimal solution and derive a well-structured solution.



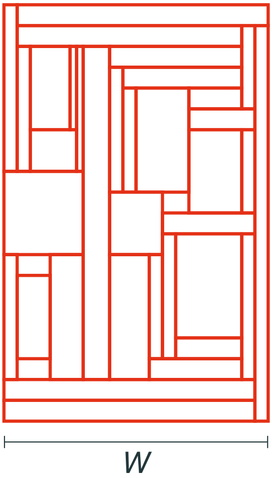
1. Consider the optimal solution and derive a well-structured solution.

Harren et al.'s algorithm



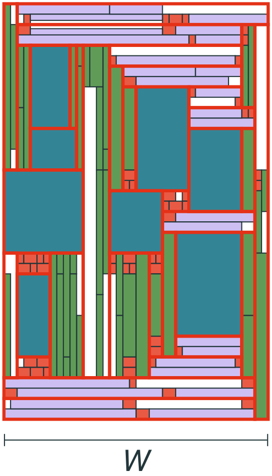
1. Consider the optimal solution and derive a well-structured solution.

Harren et al.'s algorithm



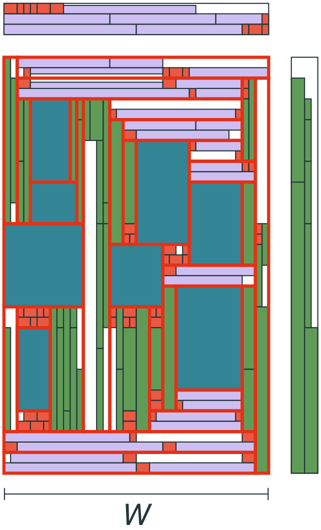
1. Consider the optimal solution and derive a well-structured solution.
2. Iterate all structures to find the correct one

Harren et al.'s algorithm



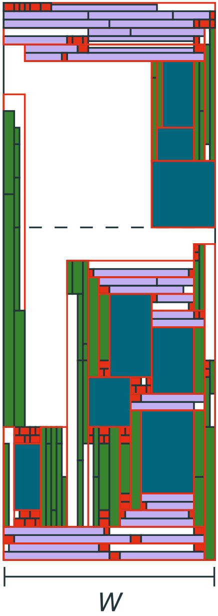
1. Consider the optimal solution and derive a well-structured solution.
2. Iterate all structures to find the correct one
3. Add almost all the items via linear programming

Harren et al.'s algorithm



1. Consider the optimal solution and derive a well-structured solution.
2. Iterate all structures to find the correct one
3. Add almost all the items via linear programming
4. Create two extra thin regions for the residual rectangles.

Harren et al.'s algorithm



1. Consider the optimal solution and derive a well-structured solution.
2. Iterate all structures to find the correct one
3. Add almost all the items via linear programming
4. Create two extra thin regions for the residual rectangles.
   These containers can be included by using extra $2/3 \cdot \mathrm{OPT}$ height.

Strategy

Idea

Place **all** items with height larger than $\frac{1}{2}\mathrm{OPT}$ inside the structured solution.

Idea

Place **all** items with height larger than $\frac{1}{2}\mathrm{OPT}$ inside the structured solution. This way the additional container only needs a height of at most $\frac{1}{2}\mathrm{OPT}$.

Strategy

Idea

Place **all** items with height larger than $\frac{1}{2}$OPT inside the structured solution. This way the additional container only needs a height of at most $\frac{1}{2}$OPT.

Structural lemma

If $\mathcal{I}$ is an instance of δ-skewed Strip Packing, there exists a well-structured solution of height at most $\left(\frac{3}{2} + \varepsilon\right) OPT$ for $\mathcal{I}$ such that:

- The rectangles of height larger than $\frac{1}{2}OPT$ can be packed exactly in polynomial time into the solution,

Strategy

Idea

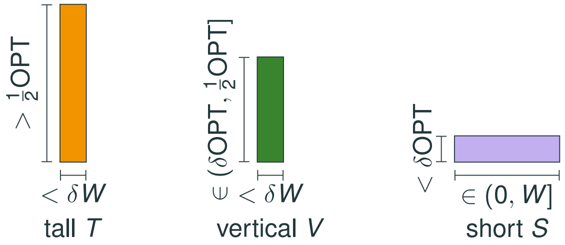
Place **all** items with height larger than $\frac{1}{2}$OPT inside the structured solution. This way the additional container only needs a height of at most $\frac{1}{2}$OPT.

Structural lemma

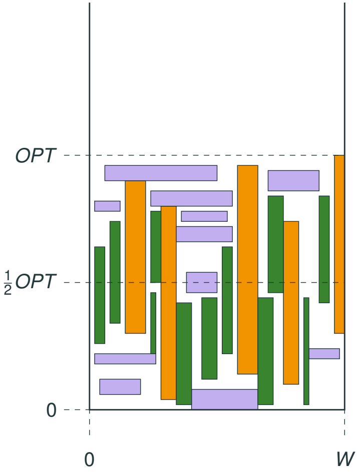
If $\mathcal{I}$ is an instance of δ-skewed Strip Packing, there exists a well-structured solution of height at most $\left(\frac{3}{2} + \varepsilon\right)$ *OPT* for $\mathcal{I}$ such that:

- The rectangles of height larger than $\frac{1}{2}$*OPT* can be packed exactly in polynomial time into the solution, and
- There is a free rectangular region of height $\frac{1}{2}$*OPT* and width ε*W* inside the solution.

New item classification

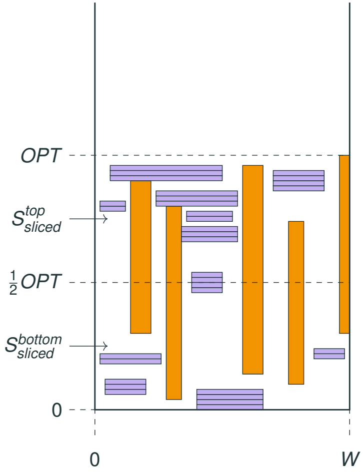


Generating the Structured Packing



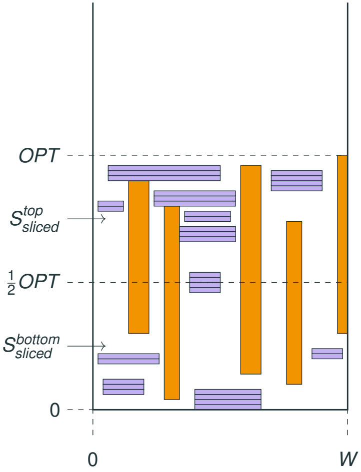
- Classification:
  - T: $h(i) > \frac{1}{2}OPT$;
  - V: $\delta OPT \leq h(i) \leq \frac{1}{2}OPT$;
  - S: $h(i) < \delta OPT$.

Generating the Structured Packing



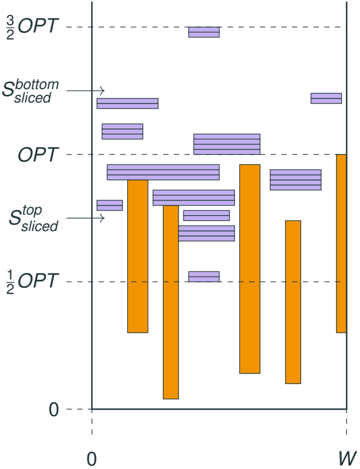
- Classification:
  - T: $h(i) > \frac{1}{2}OPT$;
  - V: $\delta OPT \leq h(i) \leq \frac{1}{2}OPT$;
  - S: $h(i) < \delta OPT$.
- Consider only T and S in OPT.

Generating the Structured Packing



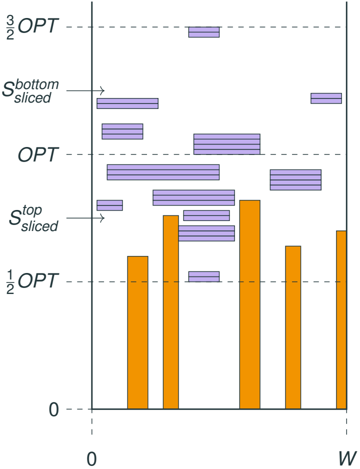
- Consider only T and S in OPT.
- Slice short items horizontally.

Generating the Structured Packing



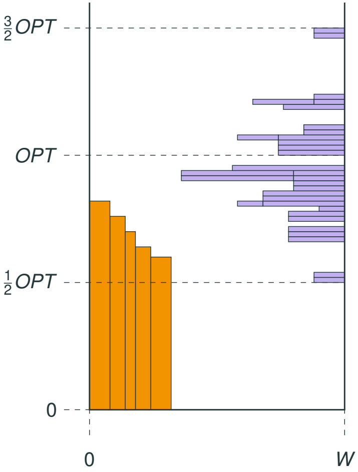
- Consider only T and S in OPT.
- Slice short items horizontally.
- Shift the bottom sliced short items up by OPT unit.

Generating the Structured Packing



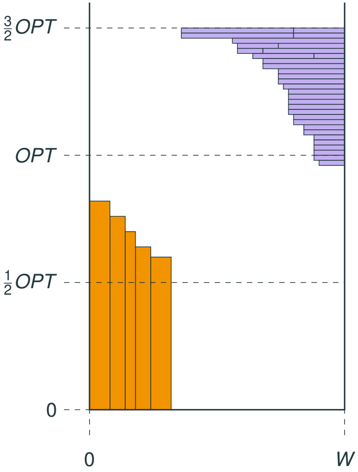
- Consider only T and S in OPT.
- Slice short items horizontally.
- Shift the bottom sliced short items up by OPT unit.
- Shift down tall items.

Generating the Structured Packing



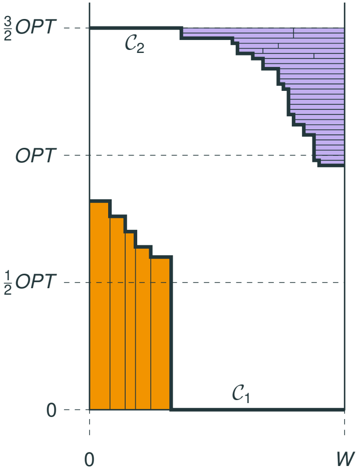
- Slice short items horizontally.
- Shift the bottom sliced short items up by OPT unit.
- Shift down tall items.
- Shift the tall items to the left and sort them according to their height, then shift the slices to the right.

Generating the Structured Packing



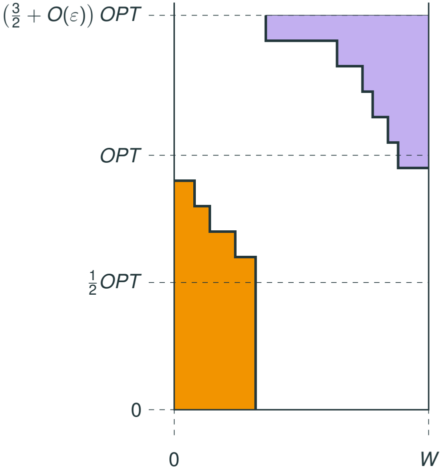
- Shift the bottom sliced short items up by OPT unit.
- Shift down tall items.
- Shift the tall items to the left and sort them according to their height, then shift the slices to the right.
- Shift up and sort stripes of short items.

Generating the Structured Packing



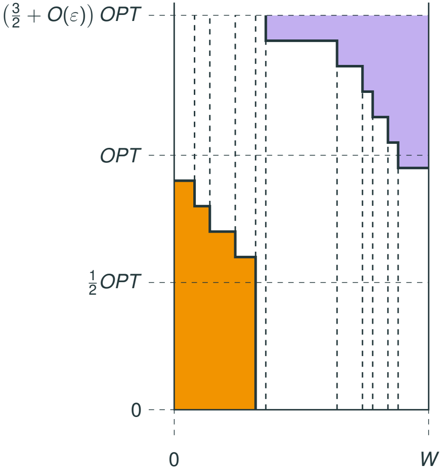
- Shift down tall items.
- Shift the tall items to the left and sort them according to their height, then shift the slices to the right.
- Shift up and sort stripes of short items.
- Silhouette of both item sets has step form

Generating the Structured Packing



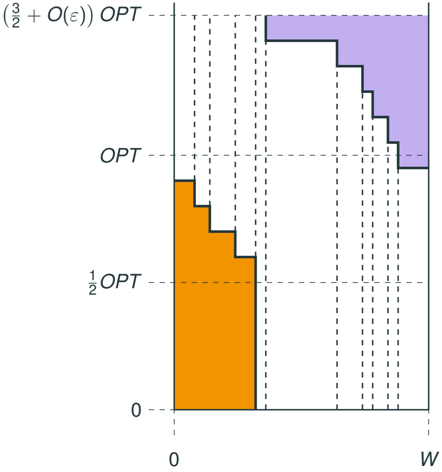
- Shift up and sort stripes of short items.
- Silhouette of both item sets has step form
- Shift the short items up by $\mathcal{O}(\varepsilon)OPT$; round up the silhouettes so as to have $O_\varepsilon(1)$ jumps.

Generating the Structured Packing



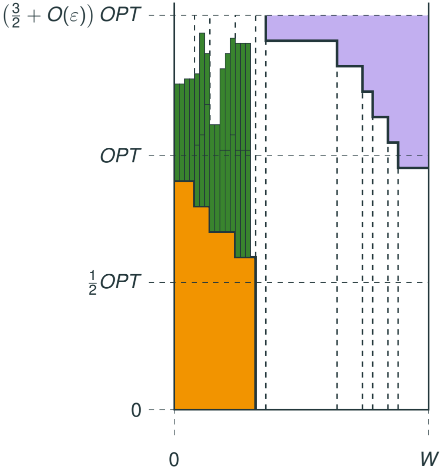
- Introduce containers for vertical items.

Generating the Structured Packing



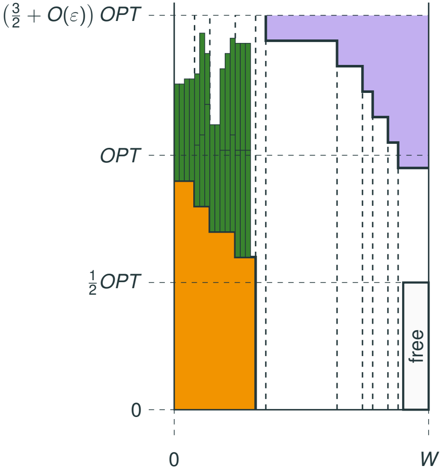
- Introduce containers for vertical items.
- Total area in containers is at least $a(V) + \left(\frac{1}{2} + \varepsilon\right) W \cdot OPT$

Generating the Structured Packing



- Introduce containers for vertical items.
- Total area in containers is at least $a(V)+\left(\frac{1}{2}+\varepsilon\right) W \cdot OPT$
- Sliced vertical items can be filled inside containers greedily because they have a height of at most $\mathrm{OPT}/2$

Generating the Structured Packing



- Introduce containers for vertical items.
- Total area in containers is at least $a(V) + \left(\frac{1}{2} + \varepsilon\right) W \cdot OPT$
- Sliced vertical items can be filled inside containers greedily because they have a height of at most OPT$/2$
- Guaranteed existence of a free area with width $\Omega(\varepsilon W)$ and height OPT$/2$

Placing items inside containers

Theorem [Kenyon and Rémila, 2000]

Given a set $\mathcal{R}$ of constantly many rectangular regions, and a set of items $\mathcal{I}$ with the following properties:

Placing items inside containers

Theorem [Kenyon and Rémila, 2000]

Given a set $\mathcal{R}$ of constantly many rectangular regions, and a set of items $\mathcal{I}$ with the following properties:

- The sliced short items fit inside the horizontal regions of $\mathcal{R}$.

Placing items inside containers

Theorem [Kenyon and Rémila, 2000]

Given a set $\mathcal{R}$ of constantly many rectangular regions, and a set of items $\mathcal{I}$ with the following properties:

- The sliced short items fit inside the horizontal regions of $\mathcal{R}$.
- The sliced vertical items fit inside the vertical regions of $\mathcal{R}$.

Placing items inside containers

Theorem [Kenyon and Rémila, 2000]

Given a set $\mathcal{R}$ of constantly many rectangular regions, and a set of items $\mathcal{I}$ with the following properties:

- The sliced short items fit inside the horizontal regions of $\mathcal{R}$.
- The sliced vertical items fit inside the vertical regions of $\mathcal{R}$.
- There exists a value $\gamma \in (0, 1)$ such that the height of each horizontal regions is at least $\frac{1}{\gamma}$ the height of the short items and the width of the vertical regions are at least $\frac{1}{\gamma}$ the width of the vertical items.

Placing items inside containers

Theorem [Kenyon and Rémila, 2000]
Given a set $\mathcal{R}$ of constantly many rectangular regions, and a set of items $\mathcal{I}$ with the following properties:

- The sliced short items fit inside the horizontal regions of $\mathcal{R}$.
- The sliced vertical items fit inside the vertical regions of $\mathcal{R}$.
- There exists a value $\gamma \in (0, 1)$ such that the height of each horizontal regions is at least $\frac{1}{\gamma}$ the height of the short items and the width of the vertical regions are at least $\frac{1}{\gamma}$ the width of the vertical items.

Then almost all items can be packed *integrally* inside the regions.

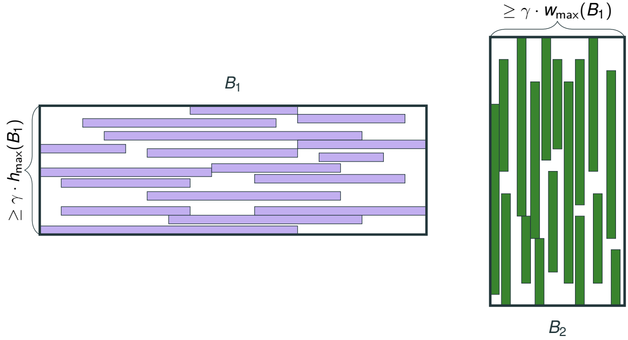
Placing items inside containers

Theorem [Kenyon and Rémila, 2000]
Given a set $\mathcal{R}$ of constantly many rectangular regions, and a set of items $\mathcal{I}$ with the following properties:

- The sliced short items fit inside the horizontal regions of $\mathcal{R}$.
- The sliced vertical items fit inside the vertical regions of $\mathcal{R}$.
- There exists a value $\gamma \in (0, 1)$ such that the height of each horizontal regions is at least $\frac{1}{\gamma}$ the height of the short items and the width of the vertical regions are at least $\frac{1}{\gamma}$ the width of the vertical items.

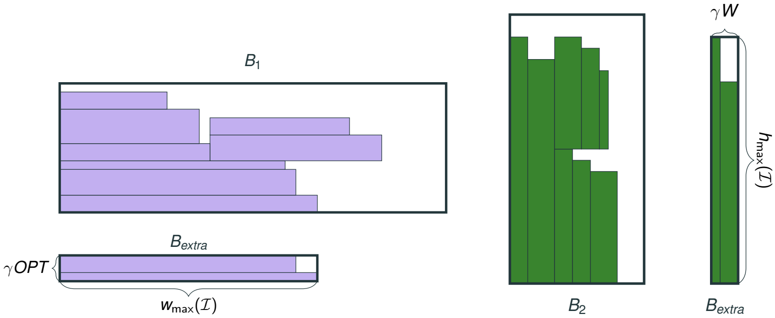
Then almost all items can be packed *integrally* inside the regions. The total area of the non-packed items can be bounded by $\mathcal{O}(\gamma) \cdot \textit{area}(\mathcal{I})$.

Placing items inside containers



Placing items inside containers

Unpacked items can be placed into thin extra boxes.



Algorithm Summary

- Place the tall items sorted by height at the bottom left.

Algorithm Summary

- Place the tall items sorted by height at the bottom left.
- Guess the containers for horizontal items, and generate the containers for vertical items.

Algorithm Summary

- Place the tall items sorted by height at the bottom left.
- Guess the containers for horizontal items, and generate the containers for vertical items.
- Use Linear Programming to place almost all horizontal and vertical items inside the containers.
  (for the correct guess they fit as sliced items)

Algorithm Summary

- Place the tall items sorted by height at the bottom left.
- Guess the containers for horizontal items, and generate the containers for vertical items.
- Use Linear Programming to place almost all horizontal and vertical items inside the containers.
  (for the correct guess they fit as sliced items)
- Place the residual items in one box with width W and height $\mathcal{O}(\varepsilon)\mathrm{OPT}$ at the top and inside the free area of height $\mathrm{OPT}/2$ and width $\mathcal{O}(\varepsilon)W$.

Summary

Theorem

For any $\delta > 0$ *and* $\varepsilon > 0$*, there is no polynomial time* $(3/2 - \varepsilon)$*-approximation for* δ*-skewed Strip Packing unless* $P = NP$*.*

Summary

Theorem

For any $\delta > 0$ *and* $\varepsilon > 0$*, there is no polynomial time* $(3/2 - \varepsilon)$*-approximation for* δ*-skewed Strip Packing unless* $P = NP$*.*

Theorem

For any $\varepsilon > 0$ *there exists* $\delta > 0$ *such that there is a* $(3/2 + \varepsilon)$*-approximation for* δ*-skewed Strip Packing.*

Open Problems

- Can we improve the approximation factor for Strip Packing from $\frac{5}{3}+\varepsilon$?

Open Problems

- Can we improve the approximation factor for Strip Packing from $\frac{5}{3} + \varepsilon$? Or improve the hardness of approximation?

Open Problems

- Can we improve the approximation factor for Strip Packing from $\frac{5}{3} + \varepsilon$? Or improve the hardness of approximation?
- For which other classes of items one can obtain tight results? For example lower bounded height or lower bounded width?

Open Problems

- Can we improve the approximation factor for Strip Packing from $\frac{5}{3} + \varepsilon$? Or improve the hardness of approximation?
- For which other classes of items one can obtain tight results? For example lower bounded height or lower bounded width?
- Strip Packing with constant number of types?

Thank You for
Your Attention