Polynomial Approximation Schemes for Smoothed and Random Instances of Multidimensional Packing Problems

Krzysztof Onak
MIT

Joint work with **David Karger** (MIT)

Bin Packing

- General framework:
  - Given is a set of items.
  - The items must be placed into bins.
  - We wish to minimize the number of bins used.
  - Versions of the problem differ on what the items are and when a subset of the items fits into a single bin.

Bin Packing

- General framework:
  - Given is a set of items.
  - The items must be placed into bins.
  - We wish to minimize the number of bins used.
  - Versions of the problem differ on what the items are and when a subset of the items fits into a single bin.

- A very natural problem that occurs in a variety of life situations.

Real-World Examples

Real-World Examples

Real-World Examples

Real-World Examples

Classical Bin Packing

Classical bin packing:

- The items are real numbers in the range $[0, 1]$.
- A set S of items fits into a single bin when

$$\sum_{a \in S} a \leq 1.$$

Multidimensional Packing Problems

Let d be a fixed number of dimensions.

Two natural generalizations of one-dimensional bin packing:

- Multidimensional bin packing (without rotations)
  - The bins are unit d-dimensional cubes.
    (For $d = 2$, the bins are squares.)
  - The items are vectors in $[0, 1]^d$, and represent d-dimensional cuboids.
    (For $d = 2$, the items are rectangles.)
  - Orientation of the items in space is fixed, and their faces are parallel to faces of bins.
  - Items must be placed inside bins so that their interiors are disjoint.

Multidimensional Packing Problems

Two natural generalizations of one-dimensional bin packing:

- Vector bin packing
  - The items are vectors in $[0,1]^d$.
  - Let S be a set of items, and let

$$(b_1, \ldots, b_d) = \sum_{a \in S} a.$$

Set S fits into a single bin if $b_i \leq 1$, for each $1 \leq i \leq d$.

Previous Work

Bin packing:

- known to be NP-hard
- Fernandez de la Vega, Lueker 1981
  - asymptotic polynomial-time approximation scheme, that is, for each $\epsilon > 0$, there is an algorithm that computes in polynomial time a packing of size

$$(1+\epsilon)\mathrm{OPT}(S) + O(1)$$

- Karmarkar, Karp 1982
  - an algorithm that produces a packing of size

$$\mathrm{OPT}(S) + O(\log^2 \mathrm{OPT}(S))$$

Previous Work

Multidimensional bin packing:

- Bansal, Sviridenko 2004
  - no asymptotic polynomial-time approximation scheme in two dimensions
- Bansal, Caprara, Sviridenko 2006
  - a 1.526-asymptotic approximation algorithm
- Karp, Luby, Marchetti-Spaccamela 1984
  - algorithms for random instances
  - each item uniformly and independently chosen from $[0,1]^d$ (works also for some other distributions)
  - the expected wasted space is $\tilde{O}(n^{(d-1)/d})$ (while $\mathbb{E}_S[\mathrm{OPT}(S)] = \Theta(n)$)

Previous Work

Vector bin packing:

- Woeginger 1997
  - no asymptotic polynomial-time approximation scheme even for 2-dimensional vector bin packing
- Chekuri, Khanna 1999
  - $O(\log d)$-approximation algorithm
- Bansal, Caprara, Sviridenko 2006
  - $(\ln d + 1 + \epsilon)$-approximation algorithm for any $\epsilon > 0$

Distributions Considered

ϱ-random instance:

- $\varrho : [0,1]^d \to [0,\infty)$ is a probability density function
- each item is drawn independently according to ϱ

Distributions Considered

ϱ-random instance:

- $\varrho : [0,1]^d \to [0,\infty)$ is a probability density function
- each item is drawn independently according to ϱ

D-smooth instance ($D \geq 1$):

- each $\varrho_i : [0,1]^d \to [0,D]$, $1 \leq i \leq n$, is a probability density function bounded by D
- the i-th item is chosen at random according to ϱ_i

Our Results

All probabilities and expectations taken over random inputs.

Our Results

All probabilities and expectations taken over random inputs.

- Multidimensional and vector bin packing:
  Fix any $\epsilon > 0$. For D-smooth instances with fixed D, there is a linear-time algorithm that finds a $(1+\epsilon)$-approximation on all but $2^{-\Omega(n)}$ fraction of inputs.

Our Results

All probabilities and expectations taken over random inputs.

- Multidimensional and vector bin packing:
  Fix any $\epsilon > 0$. For D-smooth instances with fixed D, there is a linear-time algorithm that finds a $(1+\epsilon)$-approximation on all but $2^{-\Omega(n)}$ fraction of inputs.
  (Holds for ϱ-random instances as well.)

Our Results

All probabilities and expectations taken over random inputs.

- Multidimensional and vector bin packing:
  Fix any $\epsilon > 0$. For D-smooth instances with fixed D, there is a linear-time algorithm that finds a $(1+\epsilon)$-approximation on all but $2^{-\Omega(n)}$ fraction of inputs.
  (Holds for ϱ-random instances as well.)
- Only vector bin packing: For each considered class of random instances, and for any $\epsilon > 0$, there exists an algorithm that in expected linear time computes a $(1+\epsilon)$-approximation.

Connection to Smoothed Analysis

- Spielman and Teng (2001):
  - Smoothed analysis.
  - Analysis of performance of the simplex algorithm on perturbed linear programs.
  - Result: polynomial expected running time.
  - Explains good behavior of simplex in practice: measurement errors.

Connection to Smoothed Analysis

- Spielman and Teng (2001):
  - Smoothed analysis.
  - Analysis of performance of the simplex algorithm on perturbed linear programs.
  - Result: polynomial expected running time.
  - Explains good behavior of simplex in practice: measurement errors.
- This work:
  - Also analysis of algorithms on perturbed instances.
  - We assume additive perturbation.
  - Approximate solutions instead of exact solutions.
  - Often with w.h.p. in polynomial time, instead of in expected polynomial time.

This Talk

In this talk we will sketch a proof of the following statement:

2-dimensional bin packing:

For each $D \geq 1$ and $\epsilon > 0$, there exists a linear-time algorithm that for the class of D-smooth instances computes a $(1+\epsilon)$-approximation on all but $2^{-\Omega(n)}$ fraction of inputs.

Rounding of F. de la Vega and Lueker

- One-dimensional bin packing: How to pack almost optimally items larger then some small fixed $\epsilon > 0$?

Rounding of F. de la Vega and Lueker

- One-dimensional bin packing: How to pack almost optimally items larger then some small fixed $\epsilon > 0$?
- Sort the items and split them into $O(1/\epsilon^2)$ groups of (almost) the same size.

Rounding of F. de la Vega and Lueker

- One-dimensional bin packing: How to pack almost optimally items larger then some small fixed $\epsilon > 0$?
- Sort the items and split them into $O(1/\epsilon^2)$ groups of (almost) the same size.
- Round items to the largest in each group. We get a new instance of the problem with a constant number of different item sizes.

Rounding of F. de la Vega and Lueker

- One-dimensional bin packing: How to pack almost optimally items larger then some small fixed $\epsilon > 0$?
- Sort the items and split them into $O(1/\epsilon^2)$ groups of (almost) the same size.
- Round items to the largest in each group. We get a new instance of the problem with a constant number of different item sizes.
- Find in polynomial time an optimal solution to the rounded instance of the problem (by using DP or IP).

Rounding of F. de la Vega and Lueker

- One-dimensional bin packing: How to pack almost optimally items larger then some small fixed $\epsilon > 0$?
- Sort the items and split them into $O(1/\epsilon^2)$ groups of (almost) the same size.
- Round items to the largest in each group. We get a new instance of the problem with a constant number of different item sizes.
- Find in polynomial time an optimal solution to the rounded instance of the problem (by using DP or IP).
- Replace items in the optimal packing of the rounded instance with the original smaller items.

Rounding of F. de la Vega and Lueker

- One-dimensional bin packing: How to pack almost optimally items larger then some small fixed $\epsilon > 0$?
- Sort the items and split them into $O(1/\epsilon^2)$ groups of (almost) the same size.
- Round items to the largest in each group. We get a new instance of the problem with a constant number of different item sizes.
- Find in polynomial time an optimal solution to the rounded instance of the problem (by using DP or IP).
- Replace items in the optimal packing of the rounded instance with the original smaller items.
- Can show this gives a $(1 + O(\epsilon))$-approximation to the initial problem.

2-Dimensions?

- Could apply the same approach to rectangular items of all sides larger than $\delta > 0$?

2-Dimensions?

- Could apply the same approach to rectangular items of all sides larger than $\delta > 0$?

  Not all rectangles are comparable in the sense smaller–greater, e.g., $(\frac{4}{5}, \frac{1}{5})$ and $(\frac{3}{5}, \frac{3}{5})$:-(

2-Dimensions?

- Could apply the same approach to rectangular items of all sides larger than $\delta > 0$?

  Not all rectangles are comparable in the sense smaller–greater, e.g., $(\frac{4}{5}, \frac{1}{5})$ and $(\frac{3}{5}, \frac{3}{5})$:-(

- Could decompose the set of items into chains of comparable rectangles, and conduct the rounding in each of them separately?

2-Dimensions?

- Could apply the same approach to rectangular items of all sides larger than $\delta > 0$?

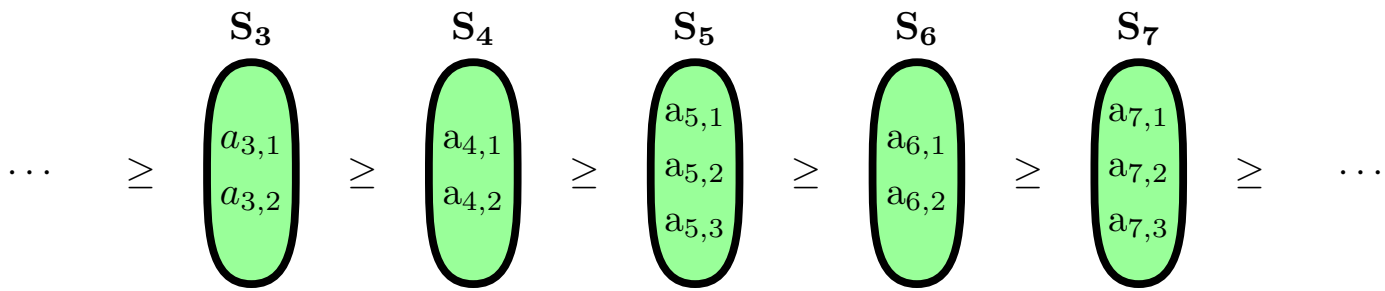
  Not all rectangles are comparable in the sense smaller–greater, e.g., $(\frac{4}{5}, \frac{1}{5})$ and $(\frac{3}{5}, \frac{3}{5})$:-(

- Could decompose the set of items into chains of comparable rectangles, and conduct the rounding in each of them separately?

  No good bound on the number of chains, and hence on the number of distinct item sizes after the rounding, even on average for most classes of random or perturbed instances :-(

Relaxing Requirements

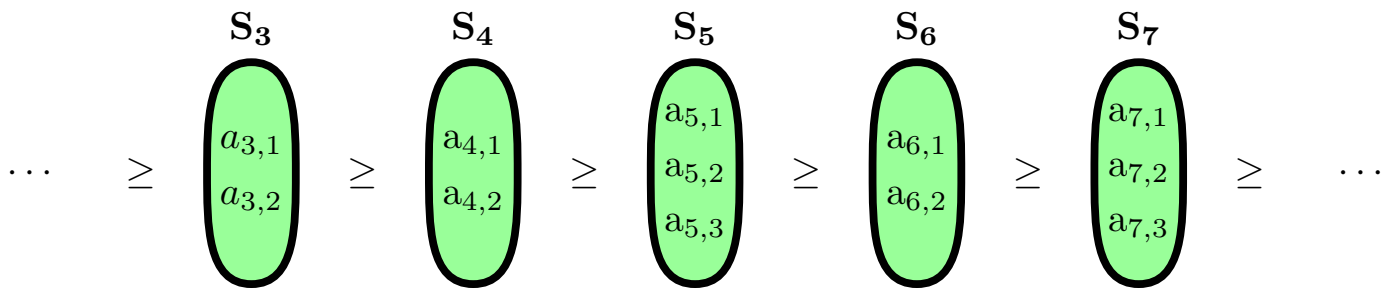
- We will conduct rounding not in chains (i.e. sets of elements $a_1 \geq a_2 \geq \ldots \geq a_k$), but in semichains, sets of sets S_i such that elements in each set S_i may not be comparable:



We call the sets S_i links.

Relaxing Requirements

- We will conduct rounding not in chains (i.e. sets of elements $a_1 \geq a_2 \geq \ldots \geq a_k$), but in semichains, sets of sets S_i such that elements in each set S_i may not be comparable:



  We call the sets S_i links.
- Can conduct rounding (that eventually gives a good approximation) if the coarseness of the semichain, defined as

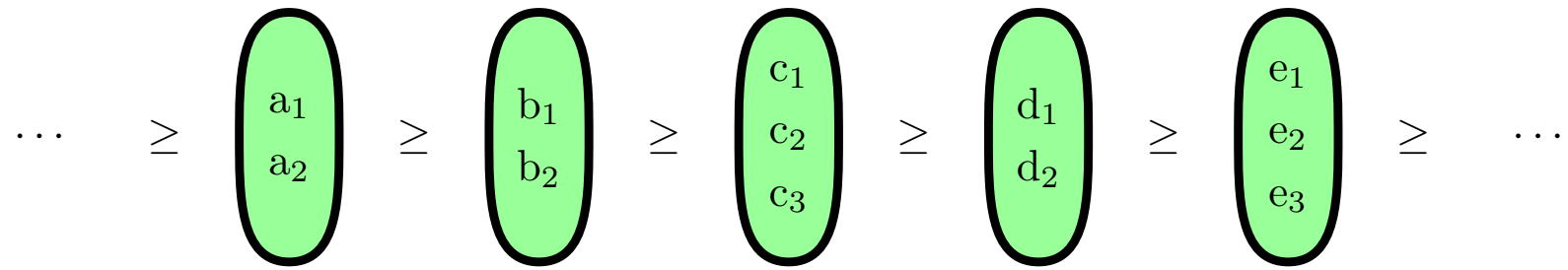
$$\frac{\max_i |S_i|}{\sum_i |S_i|},$$

  is small.

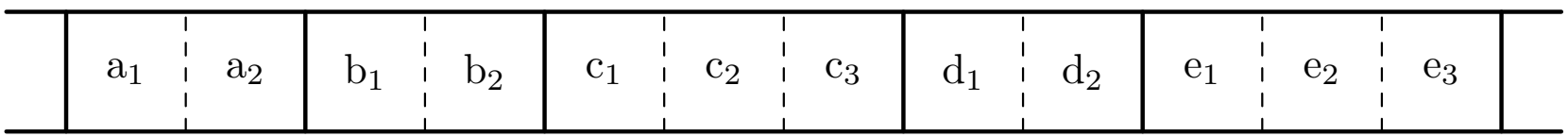
How to Round

1. Enumerate items, starting from S_1, the set of largest items to S_k, the set of smallest items.

semichain:

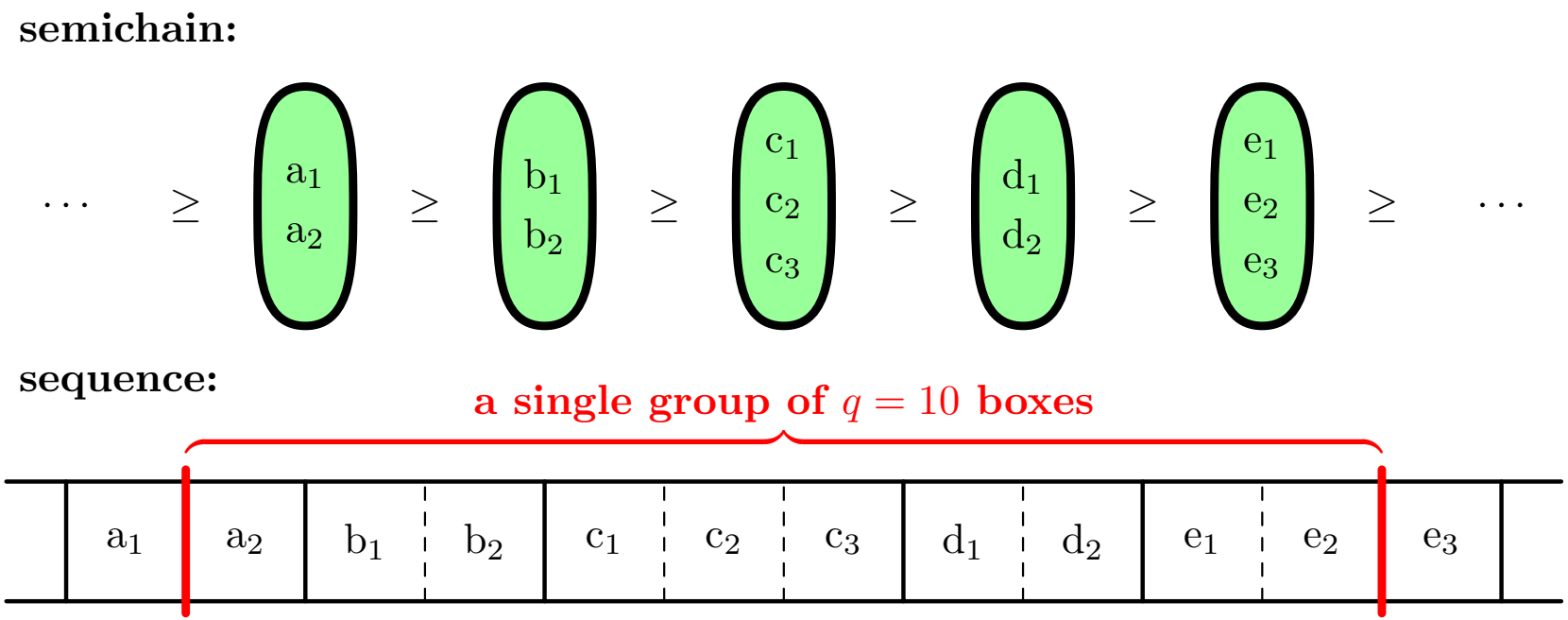


sequence:



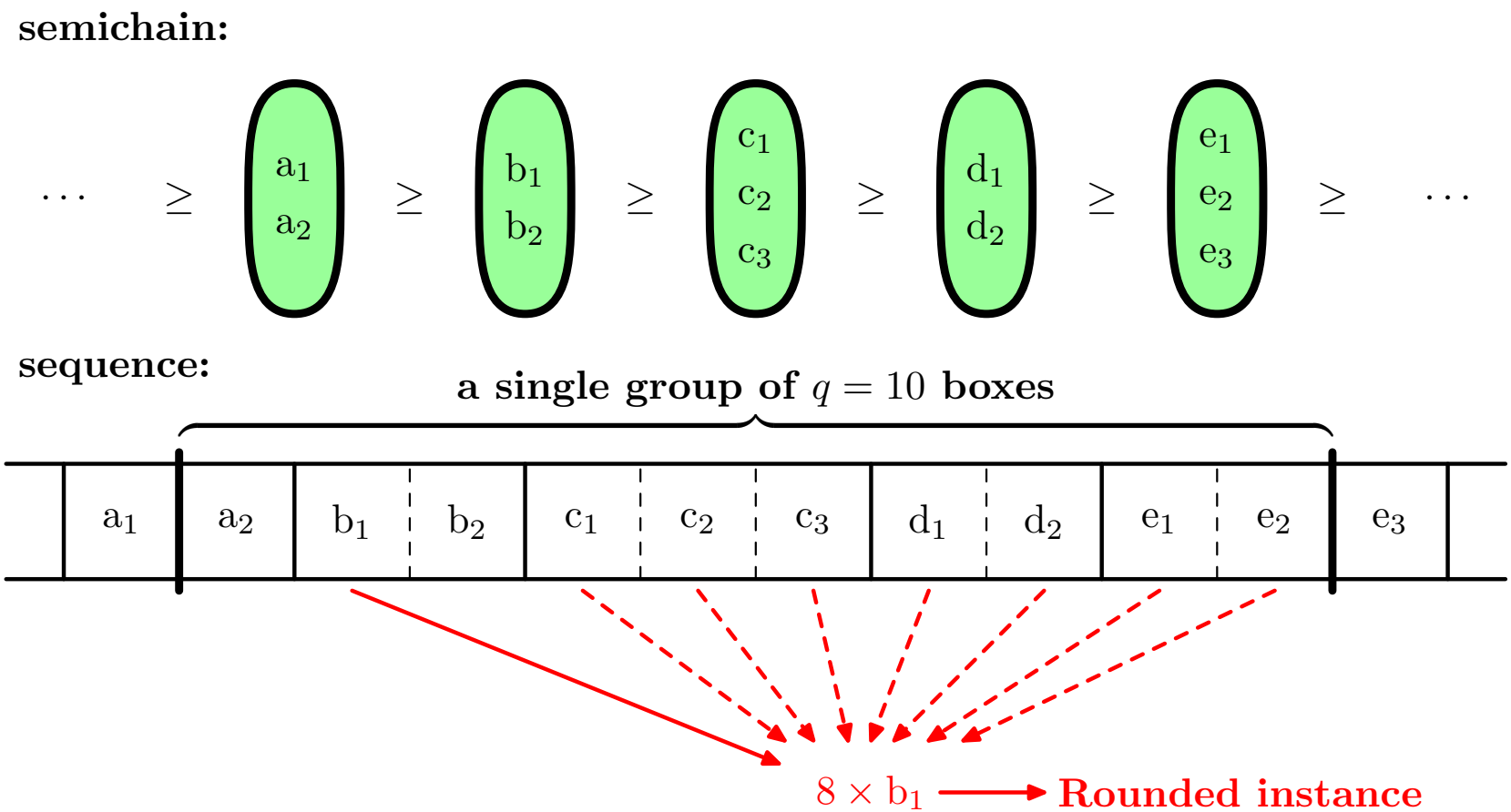
How to Round

2. Split the sequence into groups of q items.



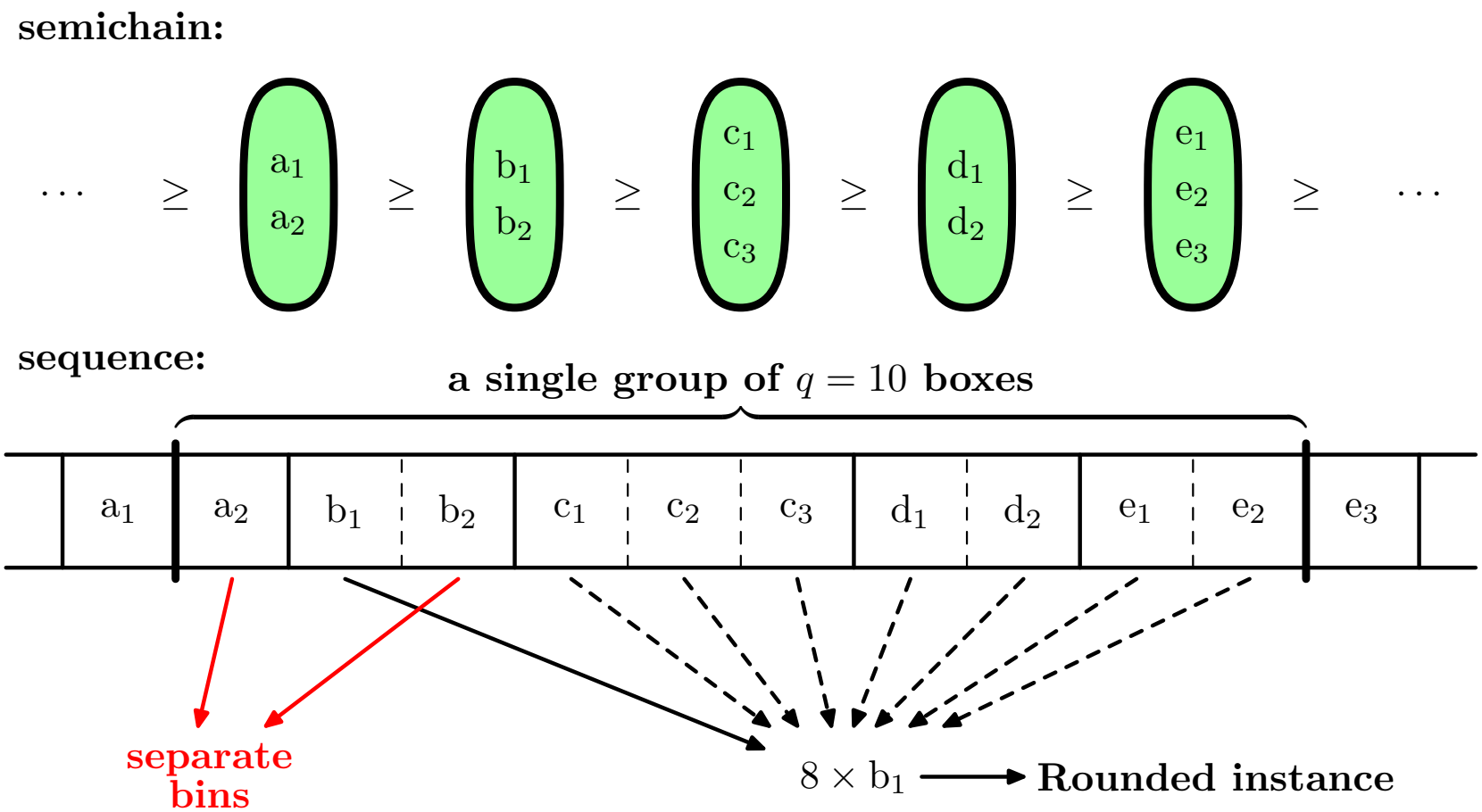
How to Round

3. For each group, pick any item that belongs to the first link S_i that occurs entirely in this group, and round all elements smaller than this element to this element.

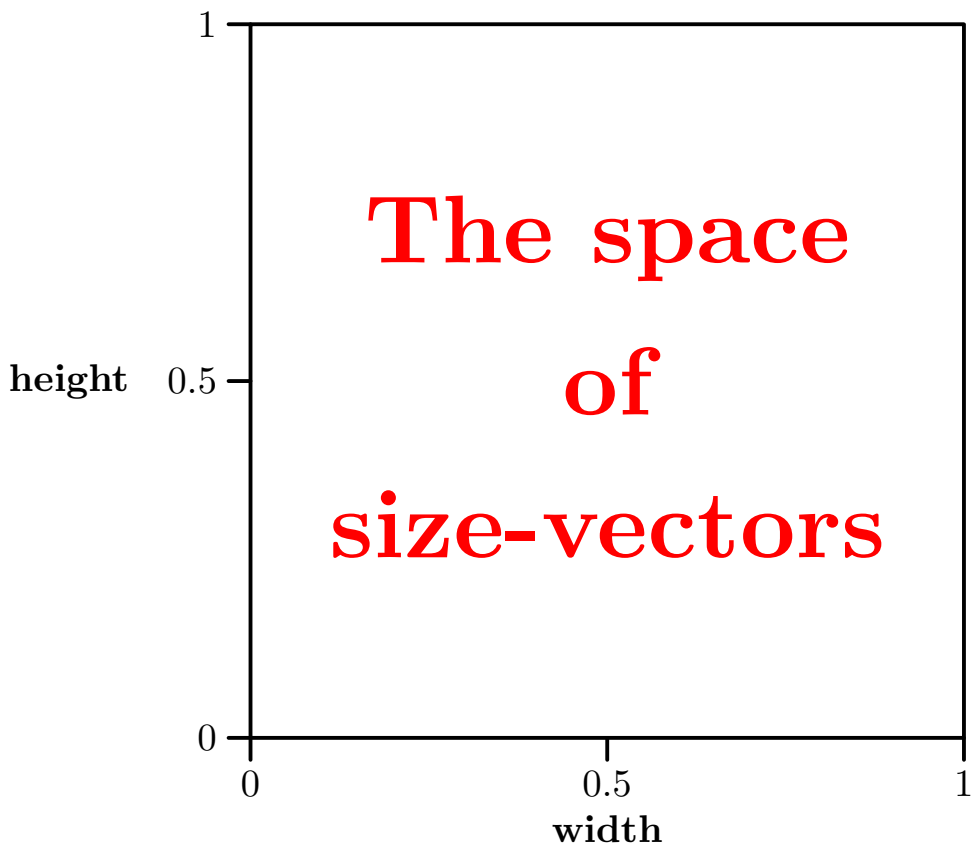


How to Round

4. Pack separately remaining items.

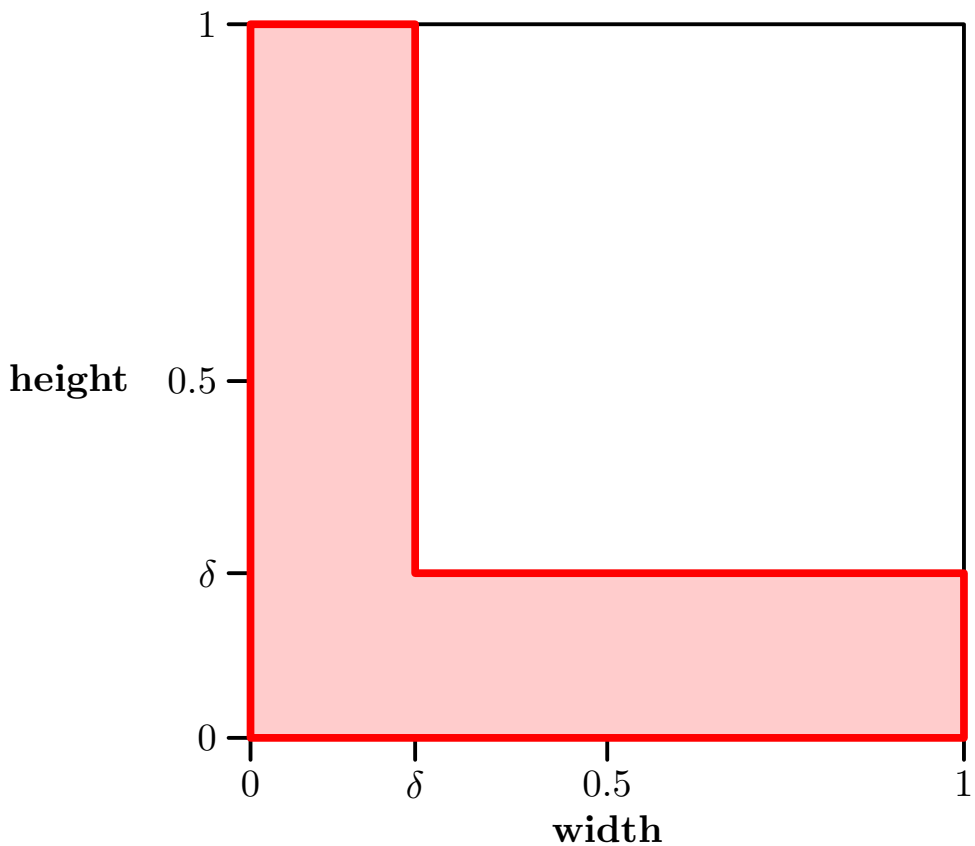


Sketch of the Algorithm



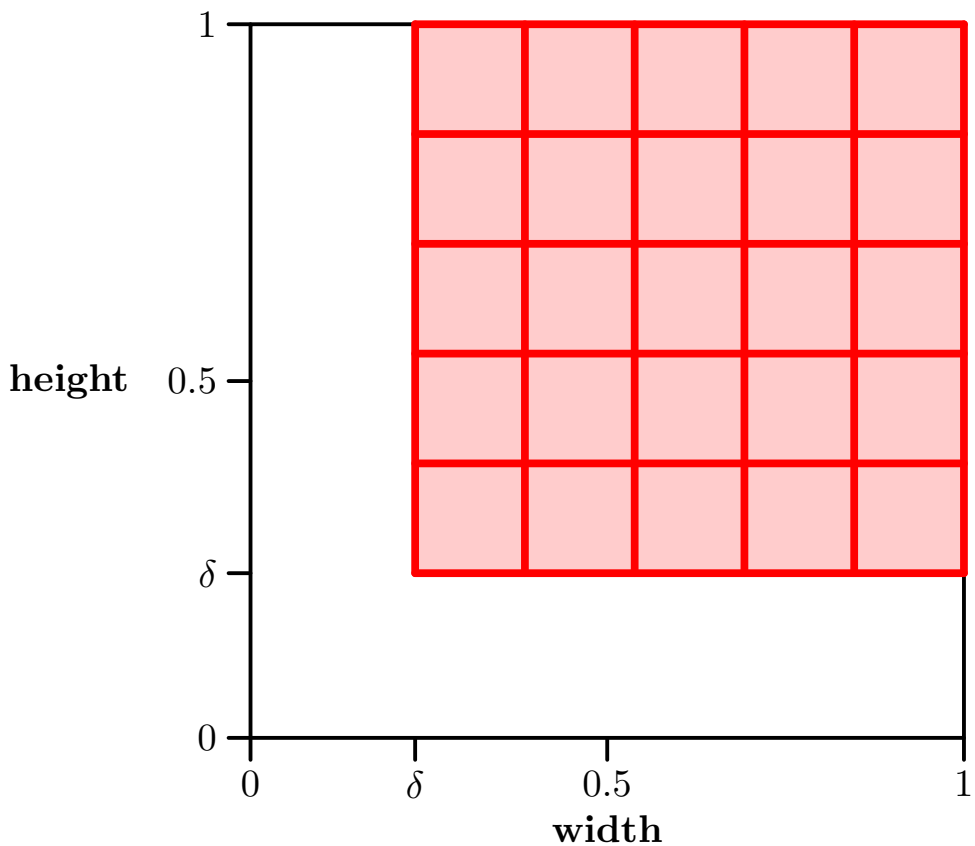
Sketch of the Algorithm

1. Pack rectangles of a side less than sufficiently small fixed δ, each into a separate bin.



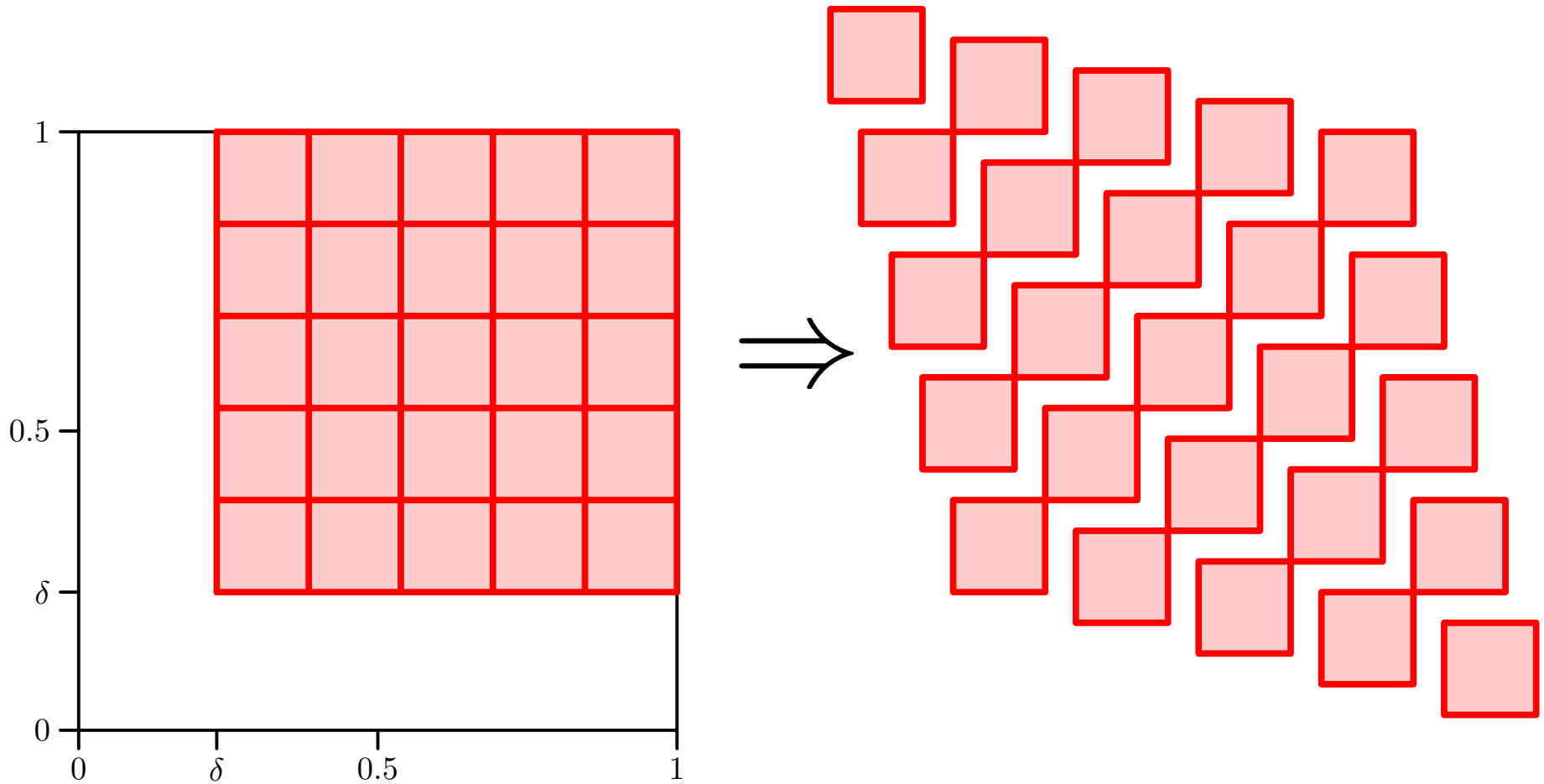
Sketch of the Algorithm

2. Cover the remaining space of size-vectors with a grid of $r \times r$ squares.



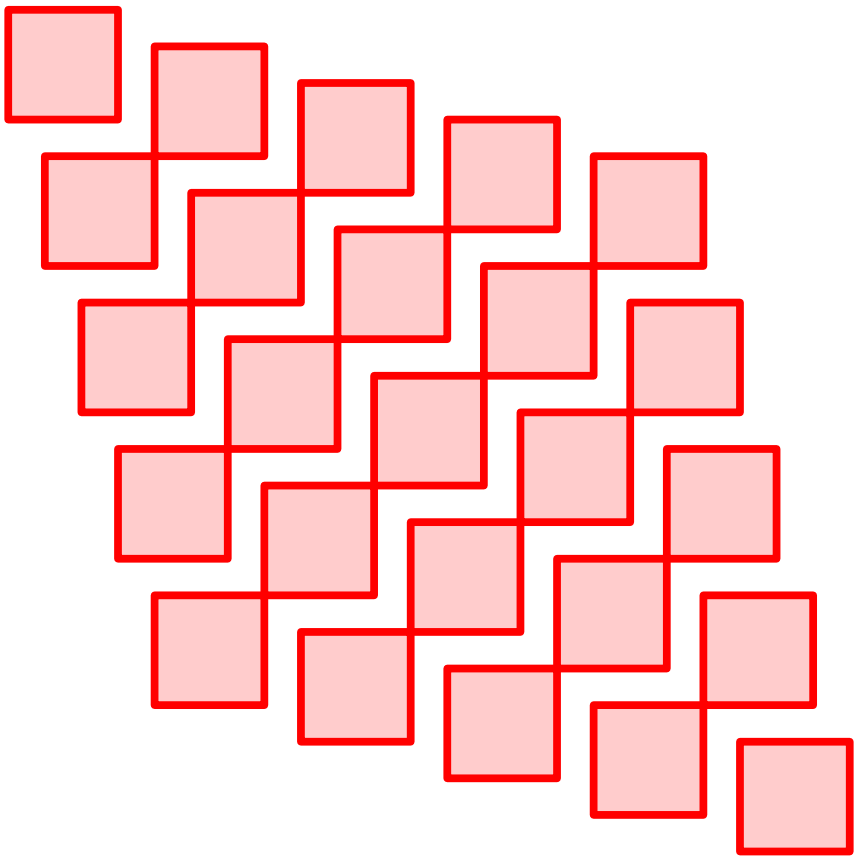
Sketch of the Algorithm

3. Vectors in each grid square constitute a link in a semichain. Group them into semichains.



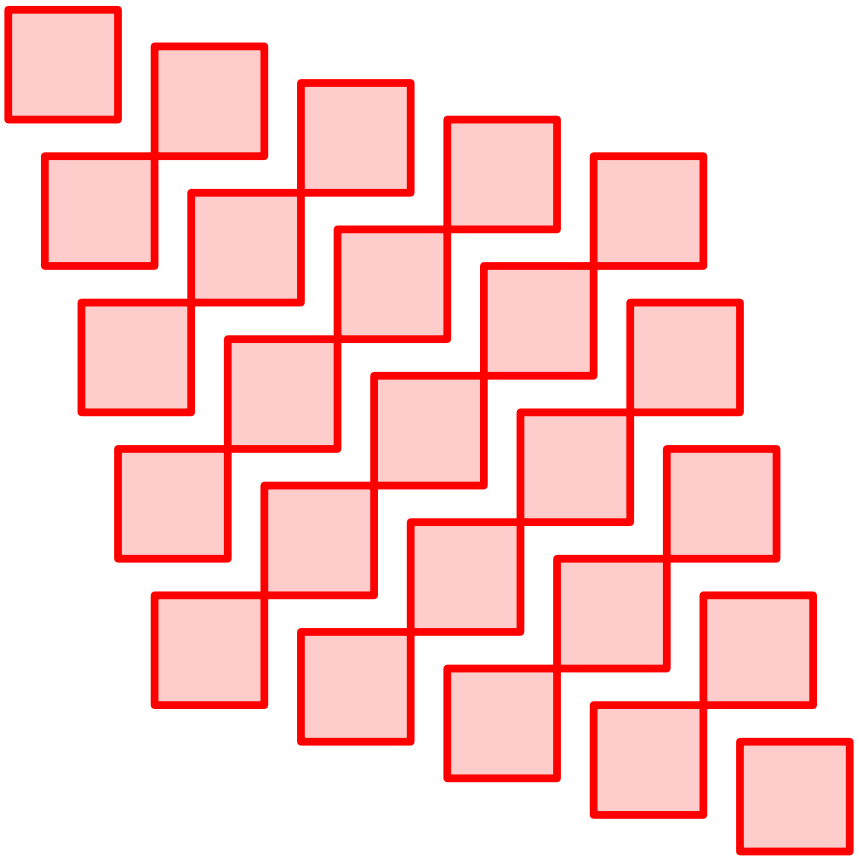
Sketch of the Algorithm

4. Drop semichains of large coarseness. Pack each item that they contain into a separate bin.



Sketch of the Algorithm

5. Pack the remaining semichains by using the multidimensional rounding.



Open Questions

- Can we show similar results for perturbation by a multiplicative rather than an additive factor?

Open Questions

- Can we show similar results for perturbation by a multiplicative rather than an additive factor?

- Are there exact algorithms of polynomial expected running time for the considered problems?

Thank you!