Streaming Algorithms
for
Bin Packing and Vector Scheduling

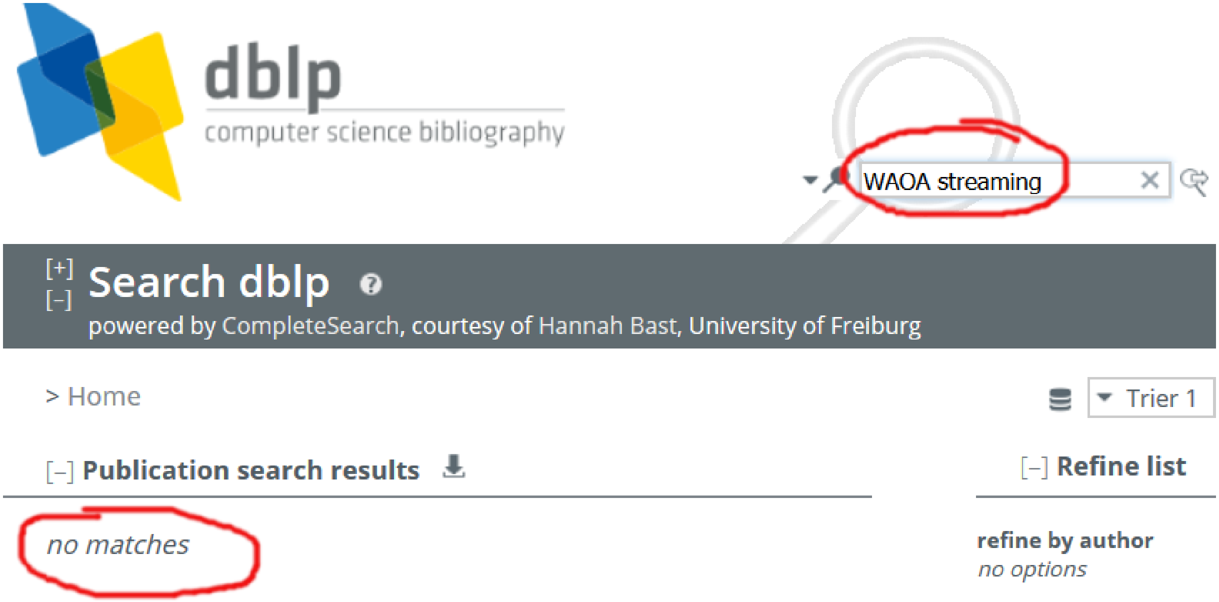
Graham Cormode and Pavel Veselý
University of Warwick

WAOA 2019, Munich

Powered by BeamerikZ

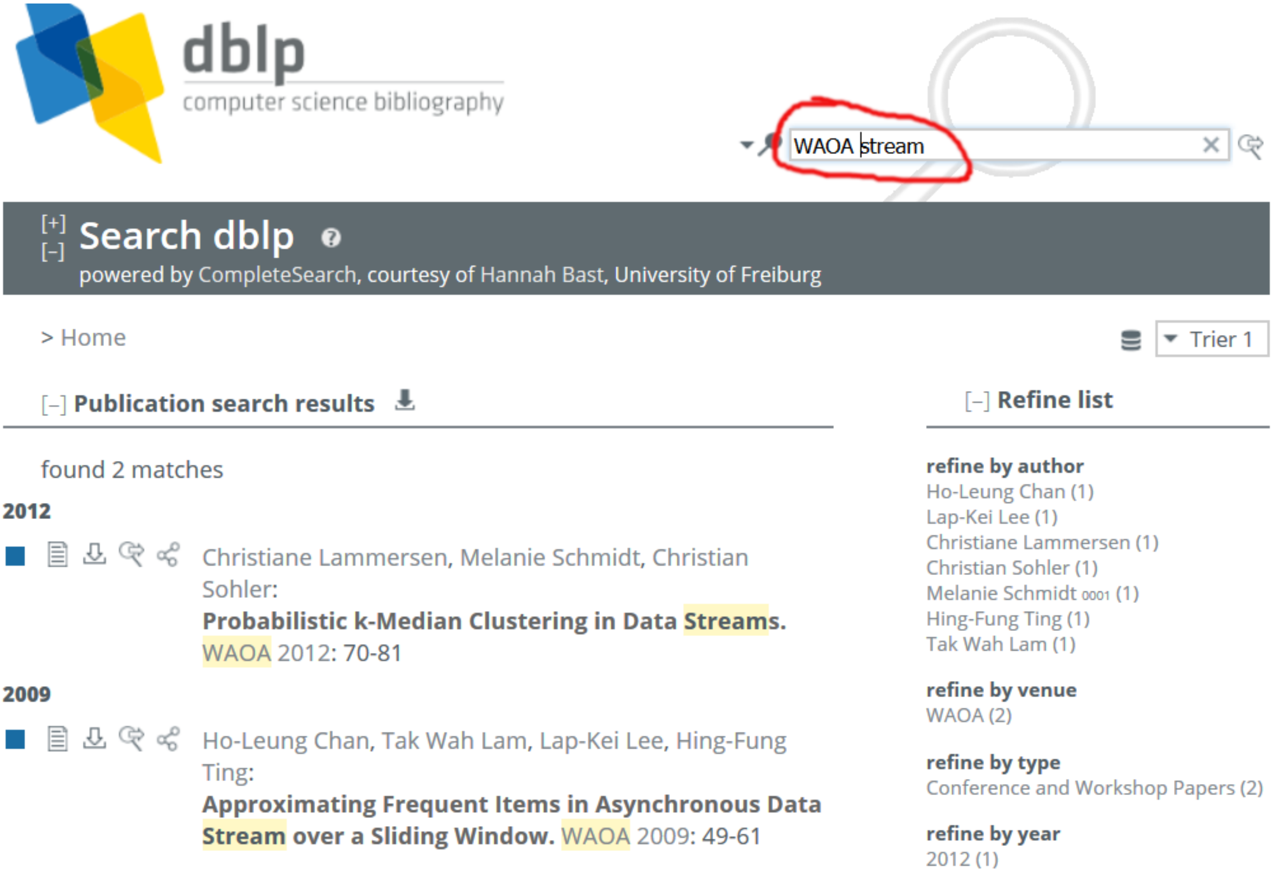
First WAOA talk containing “streaming” . . .

First WAOA talk containing "streaming" . . .

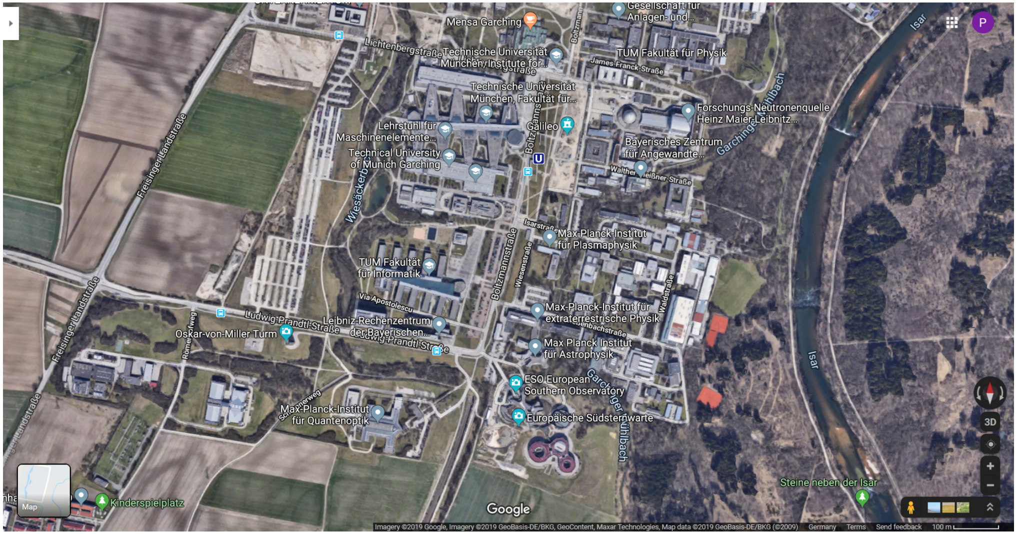


First WAOA talk containing "streaming" . . .

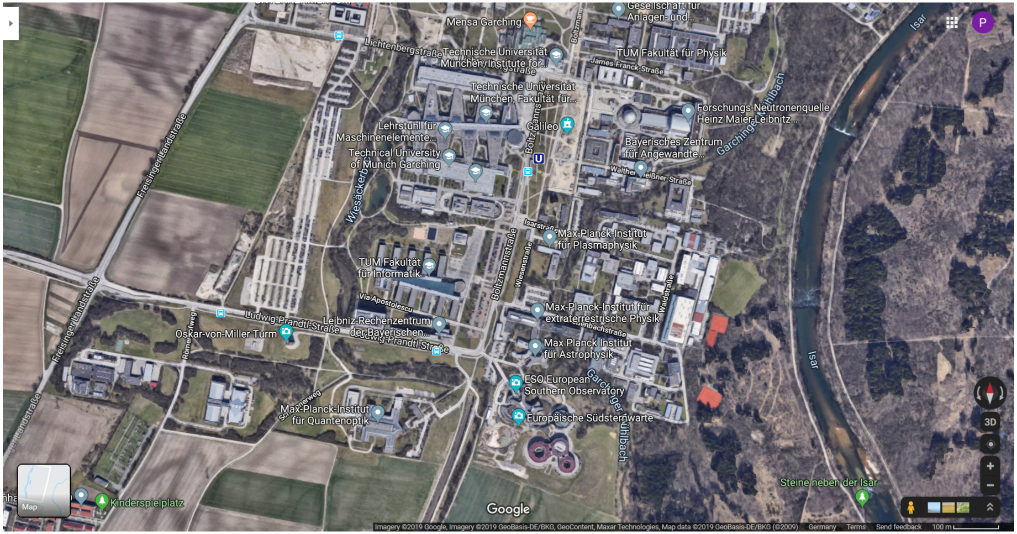
. . . but not the first one on "data streams"



Overview

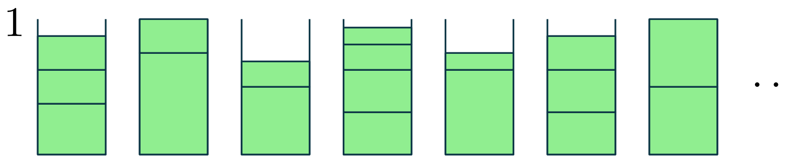

Mensa Garching
Gesellschaft für Anlagen- und...
Lichtenbergstraße
Technische Universität München, Institute for...
TUM Fakultät für Physik
James-Franck-Straße
Technische Universität München, Fakultät für...
Forschungs-Neutronenquelle Heinz Maier-Leibnitz...
Lehrstuhl für Maschinenelemente...
Galileo
Technical University of Munich Garching
Bayerisches Zentrum für Angewandte...
Freisinger Landstraße
Max-Planck-Institut für Plasmaphysik
TUM Fakultät für Informatik
Boltzmannstraße
Max-Planck-Institut für extraterrestrische Physik
Ludwig-Prandtl-Straße
Leibniz-Rechenzentrum der Bayerischen...
Oskar-von-Miller Turm
Max Planck Institut für Astrophysik
ESO European Southern Observatory
Europäische Südsternwarte
Max-Planck-Institut für Quantenoptik
Isar
Steine neben der Isar
Kinderspielplatz
Map
Google
3D

Overview

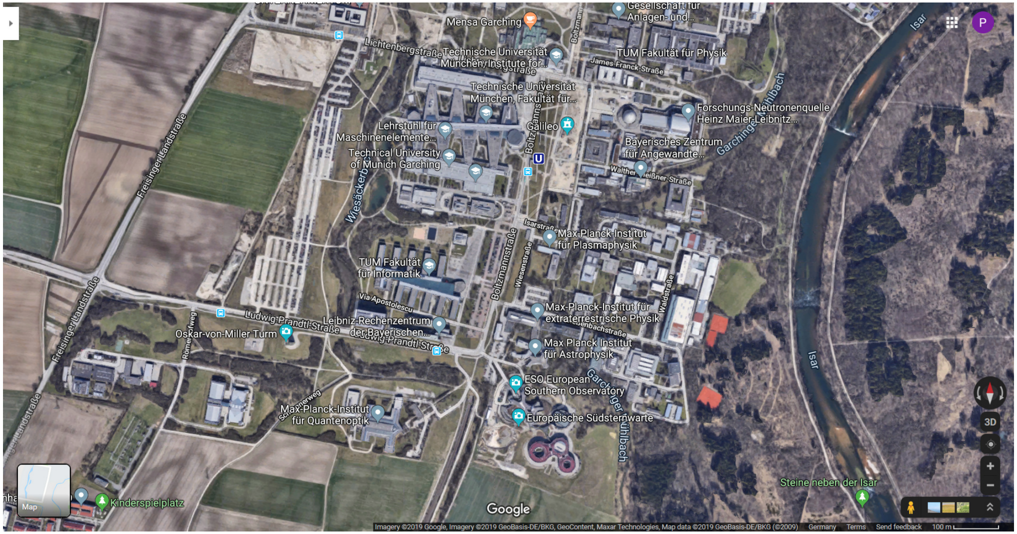


Connecting

Big Data Algorithms & **Combinatorial Optimization**

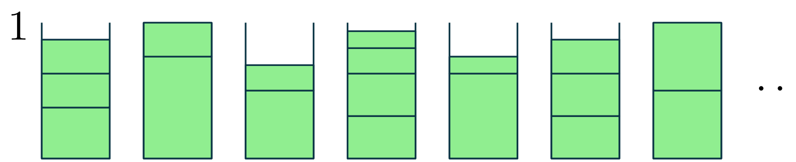


Overview



Connecting

Big Data Algorithms & **Combinatorial Optimization**



This talk's focus:

streaming algorithms | packing and scheduling

Streaming Model of Computation

Streaming Model of Computation

- One pass over data w/ limited memory

Streaming Model of Computation

- One pass over data w/ limited memory

Streaming Algorithm

- receives data in a stream, item by item
- uses memory sublinear in N = stream length

Streaming Model of Computation

- One pass over data w/ limited memory

Streaming Algorithm

- receives data in a stream, item by item
- uses memory sublinear in N = stream length
- at the end, computes approximate answer

Streaming Model of Computation

- One pass over data w/ limited memory

Streaming Algorithm

- receives data in a stream, item by item
- uses memory sublinear in N = stream length
- at the end, computes approximate answer

Note: cannot output a packing / schedule

$\Rightarrow$ estimate optimal cost (+ output template of a solution)

Streaming Model of Computation

- One pass over data w/ limited memory

Streaming Algorithm

- receives data in a stream, item by item
- uses memory sublinear in N = stream length
- at the end, computes approximate answer

Note: cannot output a packing / schedule

$\Rightarrow$ estimate optimal cost (+ output template of a solution)

Challenges:

- N very large
- Stream ordered arbitrarily
- No random access to data

Streaming Model of Computation

- One pass over data w/ limited memory

Streaming Algorithm

- receives data in a stream, item by item
- uses memory sublinear in N = stream length
- at the end, computes approximate answer

Note: cannot output a packing / schedule

$\Rightarrow$ estimate optimal cost (+ output template of a solution)

Challenges:

- N very large
- Stream ordered arbitrarily
- No random access to data

Trade-off: space vs. accuracy of the estimate

Streaming Model of Computation

- One pass over data w/ limited memory

Streaming Algorithm

- receives data in a stream, item by item
- uses memory sublinear in N = stream length
- at the end, computes approximate answer

Note: cannot output a packing / schedule

$\Rightarrow$ estimate optimal cost (+ output template of a solution)

Challenges:
- N very large
- Stream ordered arbitrarily
- No random access to data

Trade-off: space vs. accuracy of the estimate

How to summarize the input?

Streaming Model of Computation

- One pass over data w/ limited memory

Streaming Algorithm

- receives data in a stream, item by item
- uses memory sublinear in N = stream length
- at the end, computes approximate answer

Note: cannot output a packing / schedule

$\Rightarrow$ estimate optimal cost (+ output template of a solution)

Challenges:

- N very large
- Stream ordered arbitrarily
- No random access to data

$\neq$ online

no need to make online decisions about the solution

Trade-off: space vs. accuracy of the estimate

How to summarize the input?

Streaming Algorithms known for . . .

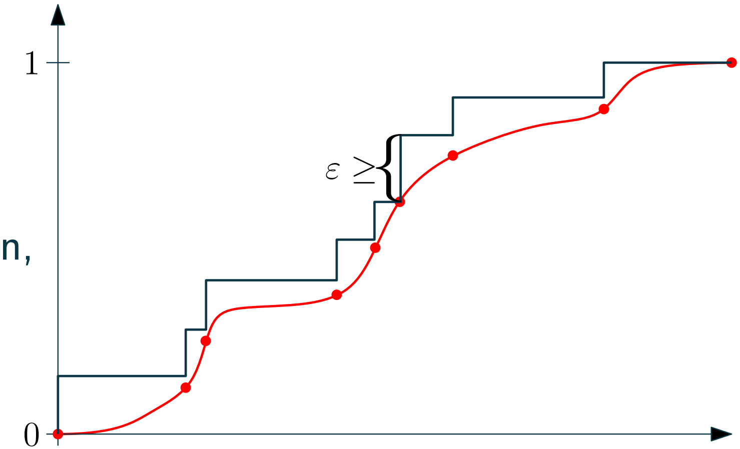
Streaming Algorithms known for . . .

- most frequent items,
- $\#$ of distinct items,

Streaming Algorithms known for ...

- most frequent items,
- $\#$ of distinct items,
- approximate median $=$ .5-quantile,
  - or any ϕ-quantile for $\phi \in [0, 1]$,
  - $= \phi \cdot N$-th largest item,
- approx. cumulative distribution function,

$$\mathrm{cdf}_A(x) = \frac{\{a \in A \,|\, a \leq x\}}{N}$$

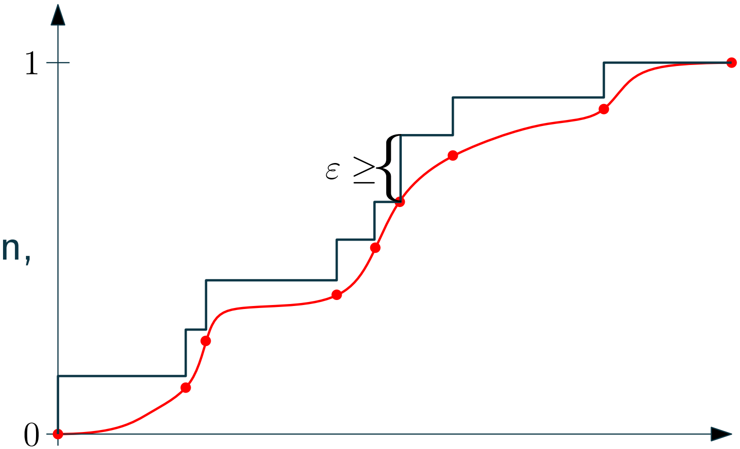


Streaming Algorithms known for ...

- most frequent items,
- $\#$ of distinct items,
- approximate median $=$ .5-quantile,
  - or any ϕ-quantile for $\phi \in [0,1]$,
  - $= \phi \cdot N$-th largest item,
- approx. cumulative distribution function,

$$\mathrm{cdf}_A(x) = \frac{\{a \in A \,|\, a \leq x\}}{N}$$

- some graph problems,
- submodular maximization,
- ...

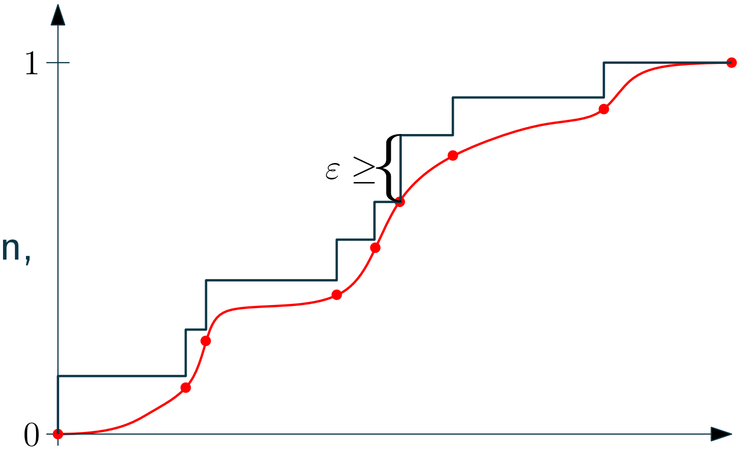


Streaming Algorithms known for ...

- most frequent items,
- $\#$ of distinct items,
- approximate median $=$ .5-quantile,
  - or any ϕ-quantile for $\phi \in [0, 1]$,
  - $= \phi \cdot N$-th largest item,
- approx. cumulative distribution function,

$$\mathrm{cdf}_A(x) = \frac{\{a \in A \,|\, a \le x\}}{N}$$

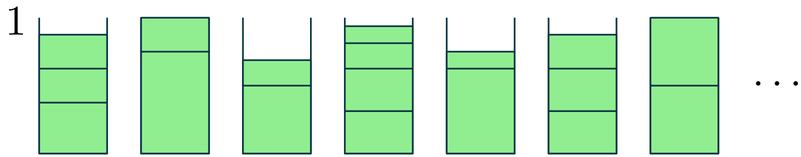
- some graph problems,
- submodular maximization,
- ...



What about other basic problems in combinatorial optimization?

Our Results: Streaming Algorithms for ...

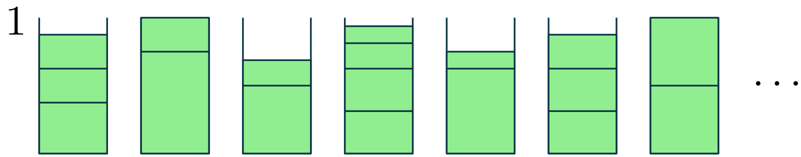
Bin Packing:



- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1
- Offline: $\mathrm{OPT} + \mathcal{O}(\log \mathrm{OPT})$ bins in poly-time [Hoberg, Rothvoss '17]

Our Results: Streaming Algorithms for ...

Bin Packing:



- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1
- Offline: $\mathrm{OPT} + \mathcal{O}(\log \mathrm{OPT})$ bins in poly-time [Hoberg, Rothvoss '17]

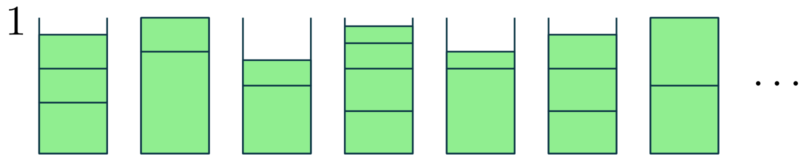
Streaming Algorithm

$1 + \varepsilon$-approximation in space $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$

Essentially best possible

Our Results: Streaming Algorithms for . . .

Bin Packing:



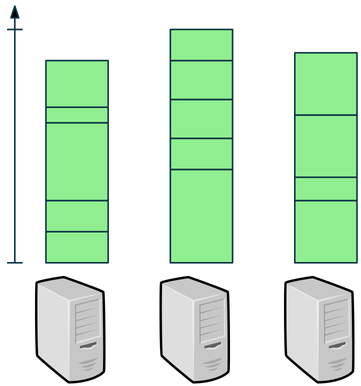
- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1
- Offline: $\mathrm{OPT} + \mathcal{O}(\log \mathrm{OPT})$ bins in poly-time [Hoberg, Rothvoss '17]

Streaming Algorithm

$1 + \varepsilon$-approximation in space $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$

Essentially best possible

Makespan Scheduling

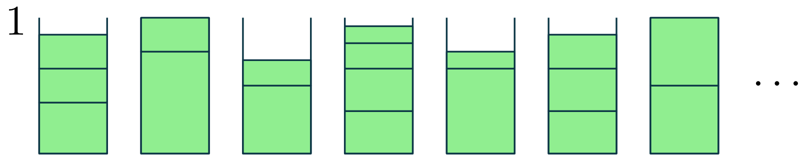


- Input: jobs with processing time
- Goal: assign jobs to machines to minimize makespan

$= $ maximum load over all machines

Our Results: Streaming Algorithms for ...

Bin Packing:

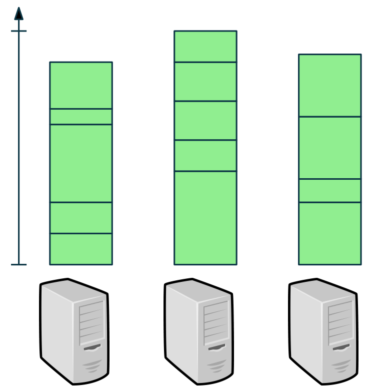


- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1
- Offline: $\mathrm{OPT} + \mathcal{O}(\log \mathrm{OPT})$ bins in poly-time [Hoberg, Rothvoss '17]

Streaming Algorithm

$1+\varepsilon$-approximation in space $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$
Essentially best possible

Makespan Scheduling



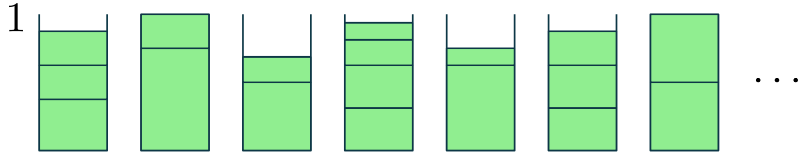
- Input: jobs with processing time
- Goal: assign jobs to machines to minimize makespan

  $=$ maximum load over all machines

- $1+\varepsilon$-approximation (rounding & DP)

Our Results: Streaming Algorithms for ...

Bin Packing:



- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1
- Offline: $\mathrm{OPT} + \mathcal{O}(\log \mathrm{OPT})$ bins in poly-time [Hoberg, Rothvoss '17]

Streaming Algorithm

$1 + \varepsilon$-approximation in space $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$

Essentially best possible

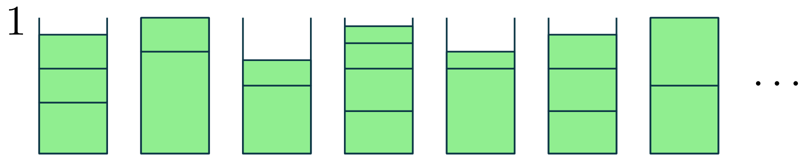
Vector Scheduling:

- Input: jobs characterized by d-dim. vectors
  - e.g.: processing time, memory or bandwidth requirements, etc.
- Goal: assign jobs to m identical machines to minimize makespan

  = maximum load over all machines and dimensions

Our Results: Streaming Algorithms for ...

Bin Packing:



- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1
- Offline: $\mathrm{OPT} + \mathcal{O}(\log \mathrm{OPT})$ bins in poly-time [Hoberg, Rothvoss '17]

Streaming Algorithm

$1 + \varepsilon$-approximation in space $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$
Essentially best possible

Vector Scheduling:

- Input: jobs characterized by d-dim. vectors
  - e.g.: processing time, memory or bandwidth requirements, etc.
- Goal: assign jobs to m identical machines to minimize makespan

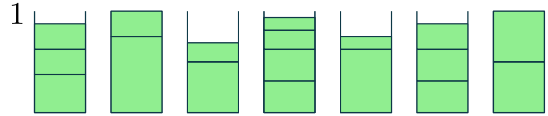
  = maximum load over all machines and dimensions

Streaming Algorithm

2-approximation in space $\widetilde{\mathcal{O}}(d^2 \cdot m^3)$

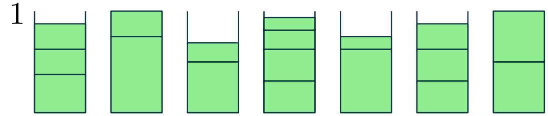
Bin Packing: Offline Approximation Scheme

- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1

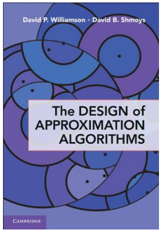


Bin Packing: Offline Approximation Scheme

- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1

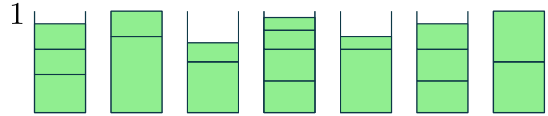


Textbook $1 + \varepsilon$-approximation from [Fernandez de la Vega, Lueker '81]

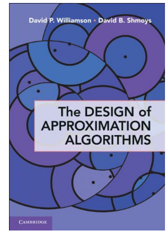


- $=$ solution with at most $(1 + \varepsilon) \cdot \mathrm{OPT} + \mathcal{O}_\varepsilon(1)$ bins

Bin Packing: Offline Approximation Scheme



- Input: items of size in $[0,1]$
- Goal: pack into min. number of bins of capacity 1

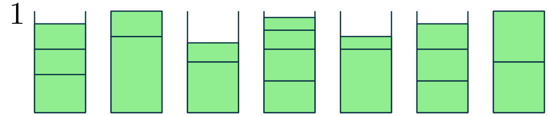


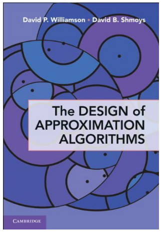
Textbook $1+\varepsilon$-approximation from [Fernandez de la Vega, Lueker '81]

- $=$ solution with at most $(1+\varepsilon)\cdot \mathrm{OPT} + \mathcal{O}_\varepsilon(1)$ bins
- Big items: size $> \varepsilon$: linear grouping

Bin Packing: Offline Approximation Scheme

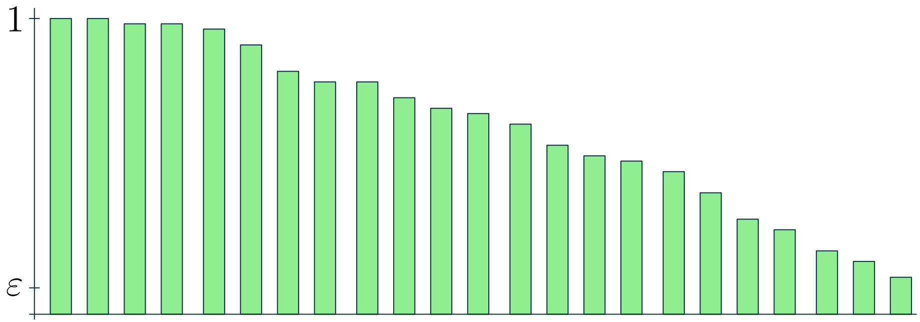
- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1





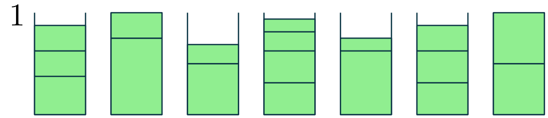
Textbook $1 + \varepsilon$-approximation from [Fernandez de la Vega, Lueker '81]

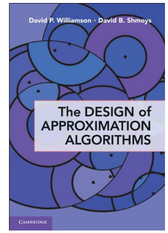
- $=$ solution with at most $(1 + \varepsilon) \cdot \mathrm{OPT} + \mathcal{O}_\varepsilon(1)$ bins
- Big items: size $> \varepsilon$: linear grouping



Bin Packing: Offline Approximation Scheme

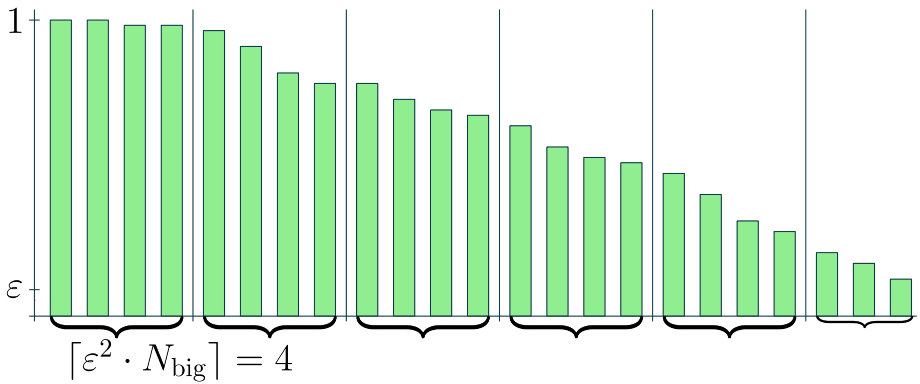
- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1





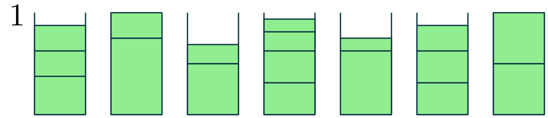
Textbook $1 + \varepsilon$-approximation from [Fernandez de la Vega, Lueker '81]

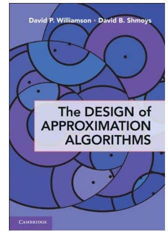
- $=$ solution with at most $(1 + \varepsilon) \cdot \mathrm{OPT} + \mathcal{O}_\varepsilon(1)$ bins
- Big items: size $> \varepsilon$: linear grouping



Bin Packing: Offline Approximation Scheme

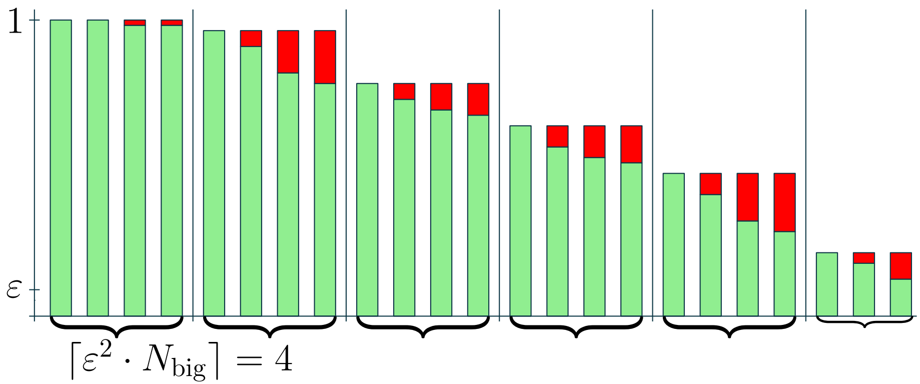
- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1





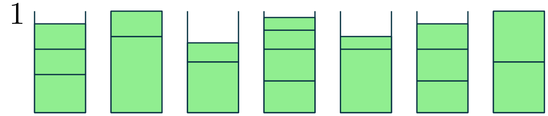
Textbook $1 + \varepsilon$-approximation from [Fernandez de la Vega, Lueker '81]

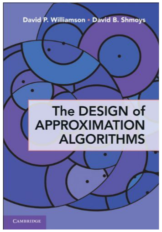
- $=$ solution with at most $(1 + \varepsilon) \cdot \mathrm{OPT} + \mathcal{O}_\varepsilon(1)$ bins
- Big items: size $> \varepsilon$: linear grouping



Bin Packing: Offline Approximation Scheme

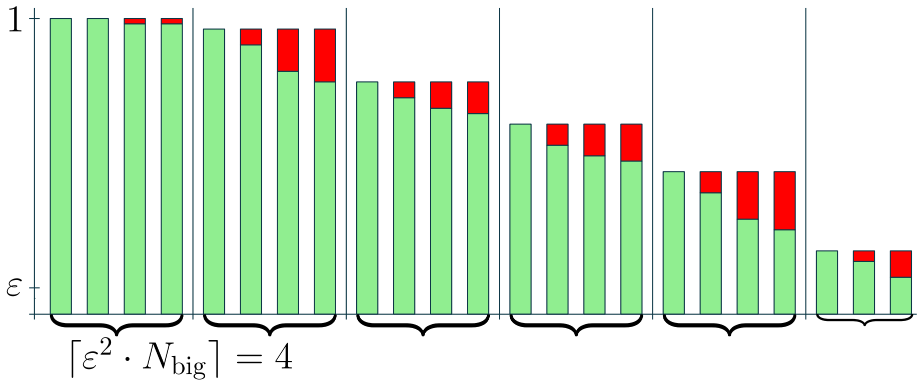
- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1





Textbook $1 + \varepsilon$-approximation from [Fernandez de la Vega, Lueker '81]

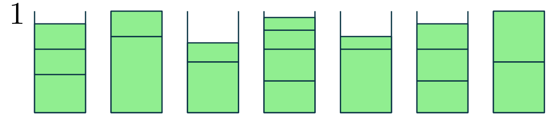
- $=$ solution with at most $(1 + \varepsilon) \cdot \mathrm{OPT} + \mathcal{O}_\varepsilon(1)$ bins
- Big items: size $> \varepsilon$: linear grouping

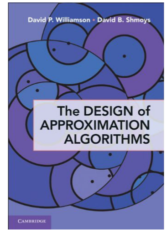


- $\rightarrow$ instance with $\lceil \frac{1}{\varepsilon^2} \rceil$ item sizes only $\rightarrow$ solve (nearly) optimally

Bin Packing: Offline Approximation Scheme

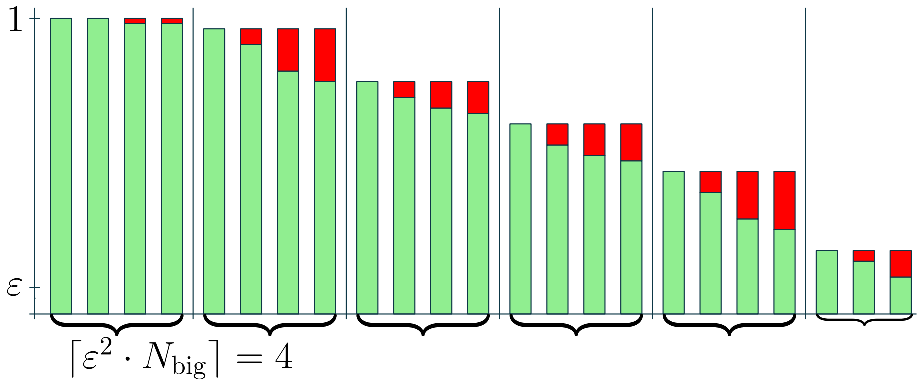
- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1





Textbook $1 + \varepsilon$-approximation from [Fernandez de la Vega, Lueker '81]

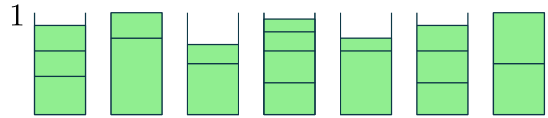
- $=$ solution with at most $(1 + \varepsilon) \cdot \mathrm{OPT} + \mathcal{O}_\varepsilon(1)$ bins
- Big items: size $> \varepsilon$: linear grouping



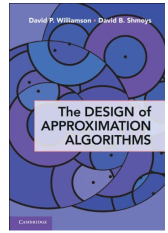
  - $\to$ instance with $\left\lceil \frac{1}{\varepsilon^2} \right\rceil$ item sizes only $\to$ solve (nearly) optimally
- Small items: size $\leq \varepsilon$:
  - Fill in greedily

Bin Packing: Offline Approximation Scheme

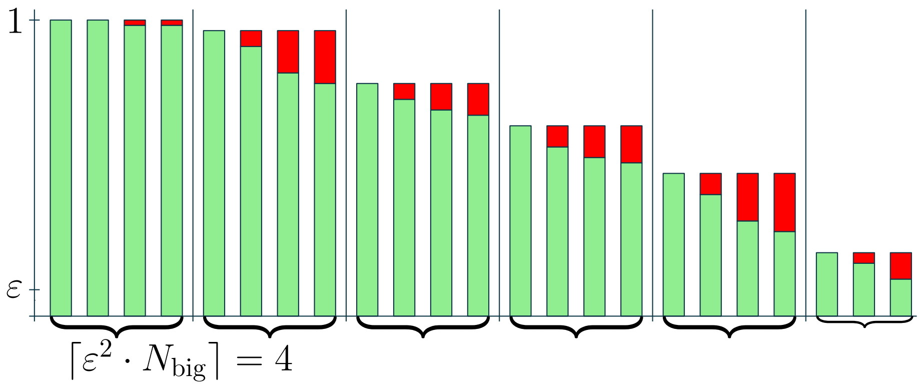
- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1



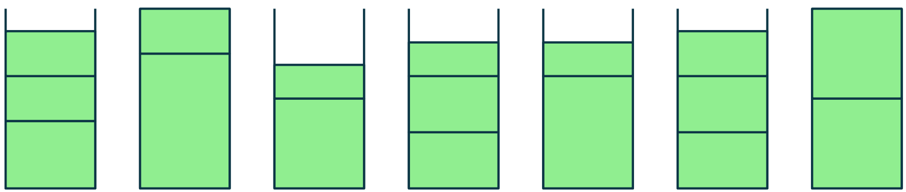
Textbook $1+\varepsilon$-approximation from [Fernandez de la Vega, Lueker '81]



- $=$ solution with at most $(1+\varepsilon)\cdot \mathrm{OPT} + \mathcal{O}_\varepsilon(1)$ bins
- Big items: size $> \varepsilon$: linear grouping

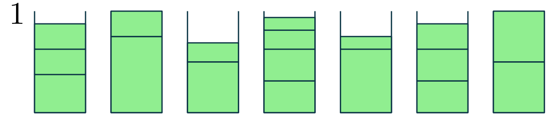


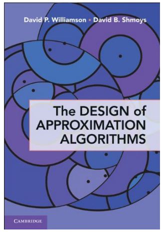
  - $\rightarrow$ instance with $\lceil \frac{1}{\varepsilon^2} \rceil$ item sizes only $\rightarrow$ solve (nearly) optimally
- Small items: size $\leq \varepsilon$:
  - Fill in greedily



Bin Packing: Offline Approximation Scheme

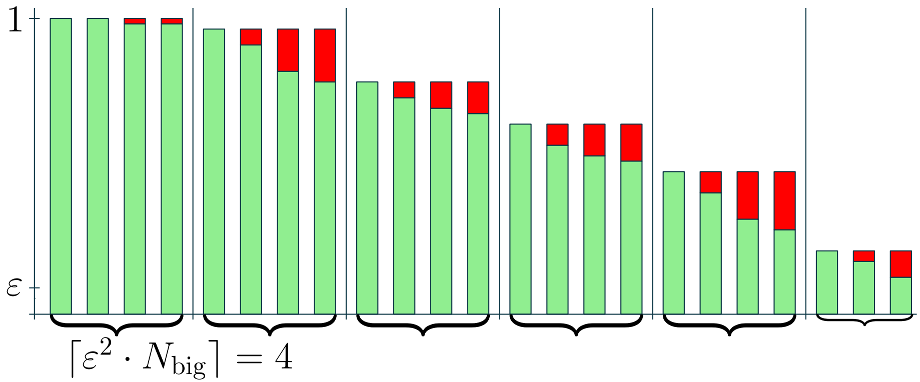
- Input: items of size in $[0,1]$
- Goal: pack into min. number of bins of capacity 1



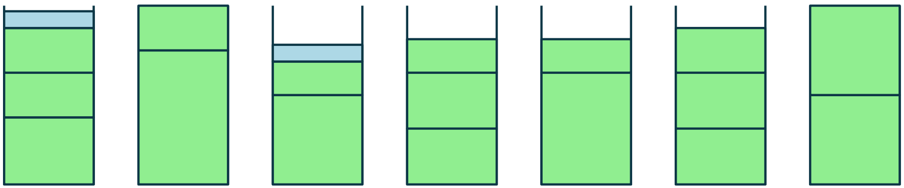


Textbook $1+\varepsilon$-approximation from [Fernandez de la Vega, Lueker '81]

- = solution with at most $(1+\varepsilon)\cdot \mathrm{OPT} + \mathcal{O}_\varepsilon(1)$ bins
- Big items: size $> \varepsilon$: linear grouping

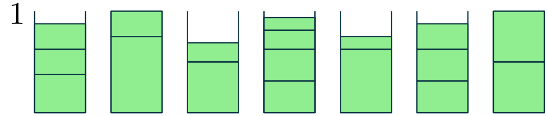


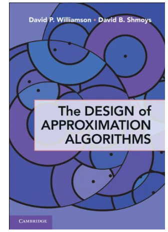
  - $\rightarrow$ instance with $\lceil \frac{1}{\varepsilon^2} \rceil$ item sizes only $\rightarrow$ solve (nearly) optimally
- Small items: size $\le \varepsilon$:
  - Fill in greedily



Bin Packing: Offline Approximation Scheme

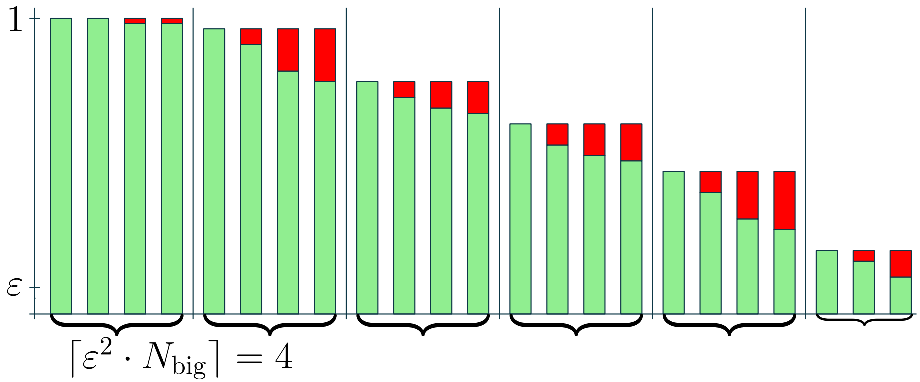
- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1



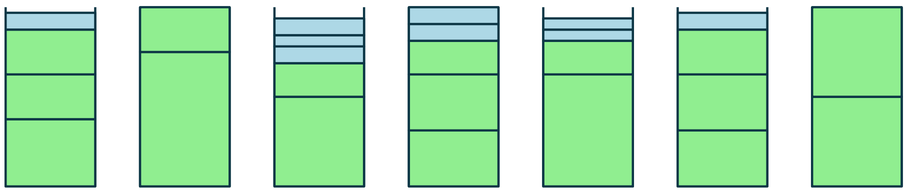


Textbook $1+\varepsilon$-approximation from [Fernandez de la Vega, Lueker '81]

- $=$ solution with at most $(1+\varepsilon)\cdot \mathrm{OPT} + \mathcal{O}_\varepsilon(1)$ bins
- Big items: size $> \varepsilon$: linear grouping

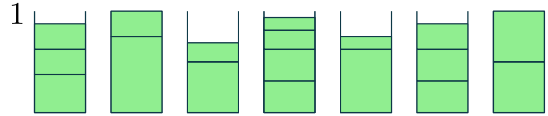


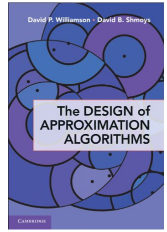
  - $\to$ instance with $\lceil \frac{1}{\varepsilon^2} \rceil$ item sizes only $\to$ solve (nearly) optimally
- Small items: size $\leq \varepsilon$:
  - Fill in greedily



Bin Packing: Offline Approximation Scheme

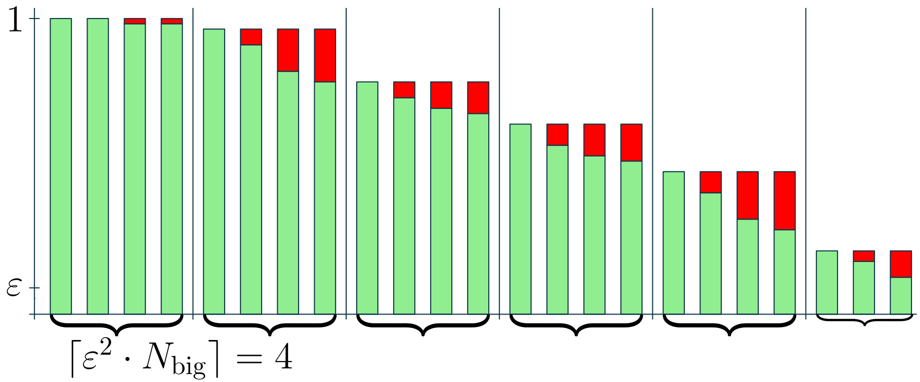
- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1



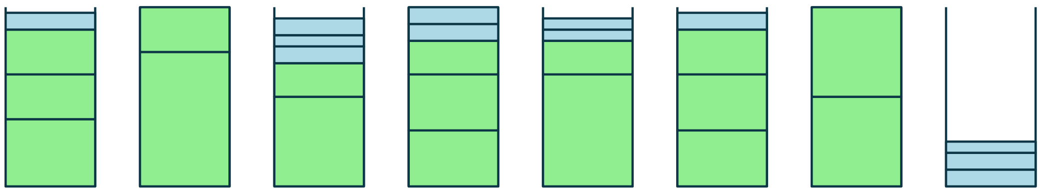


Textbook $1 + \varepsilon$-approximation from [Fernandez de la Vega, Lueker '81]

- $=$ solution with at most $(1 + \varepsilon) \cdot \mathrm{OPT} + \mathcal{O}_\varepsilon(1)$ bins
- Big items: size $> \varepsilon$: linear grouping

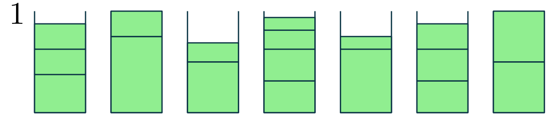


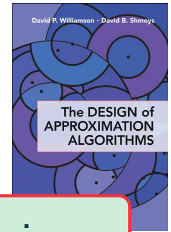
  - $\rightarrow$ instance with $\lceil \frac{1}{\varepsilon^2} \rceil$ item sizes only $\rightarrow$ solve (nearly) optimally
- Small items: size $\leq \varepsilon$:
  - Fill in greedily



Bin Packing: Offline Approximation Scheme

- Input: items of size in $[0, 1]$
- Goal: pack into min. number of bins of capacity 1

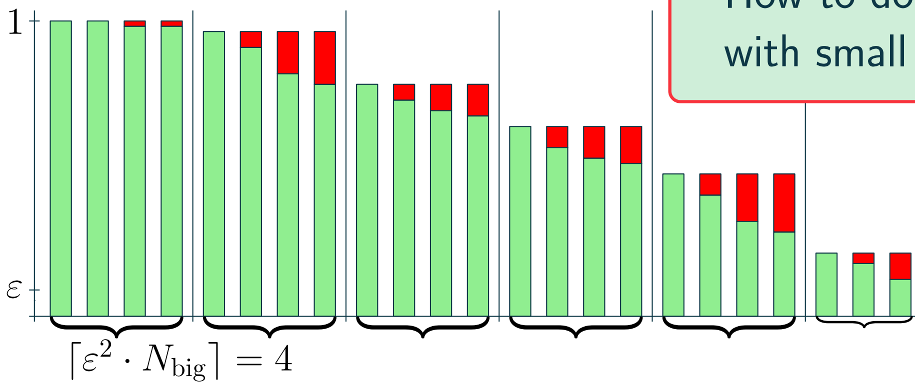




Textbook $1+\varepsilon$-approximation from [Fernandez de la Vega, Lueker '81]

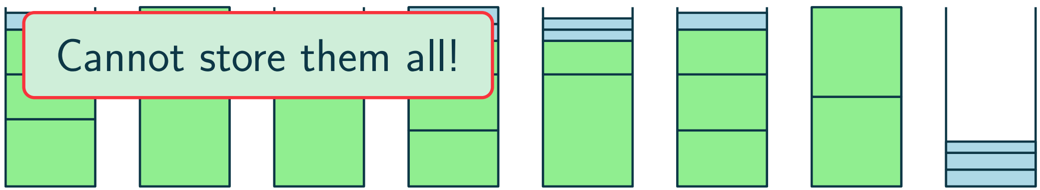
- $=$ solution with at most $(1+\varepsilon) \cdot \mathsf{OPT} + \mathcal{O}_\varepsilon(1)$ bins
- Big items: size $> \varepsilon$: linear grouping

How to do grouping with small memory?



  - $\rightarrow$ instance with $\left\lceil \frac{1}{\varepsilon^2} \right\rceil$ item sizes only $\rightarrow$ solve (nearly) optimally
- Small items: size $\leq \varepsilon$:
  - Fill in greedily

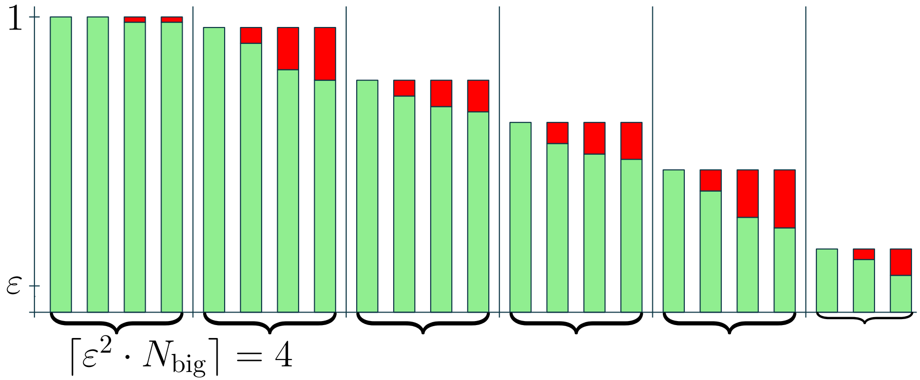
Cannot store them all!



Streaming $1 + \varepsilon$-Approx. for Bin Packing

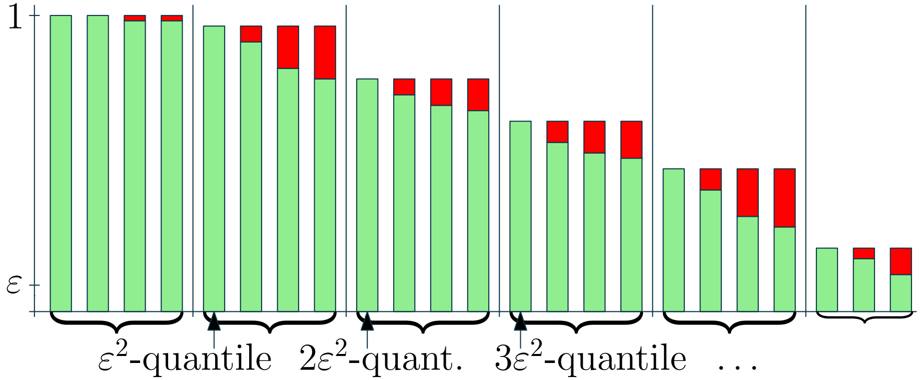
Streaming $1+\varepsilon$-Approx. for Bin Packing

Big items: size $> \varepsilon$: **approximate linear grouping**



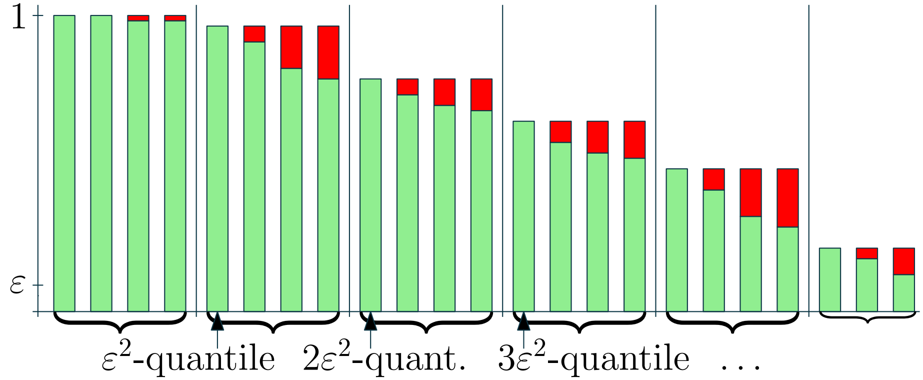
Streaming $1 + \varepsilon$-Approx. for Bin Packing

Big items: size $> \varepsilon$: **approximate linear grouping**



Streaming $1 + \varepsilon$-Approx. for Bin Packing

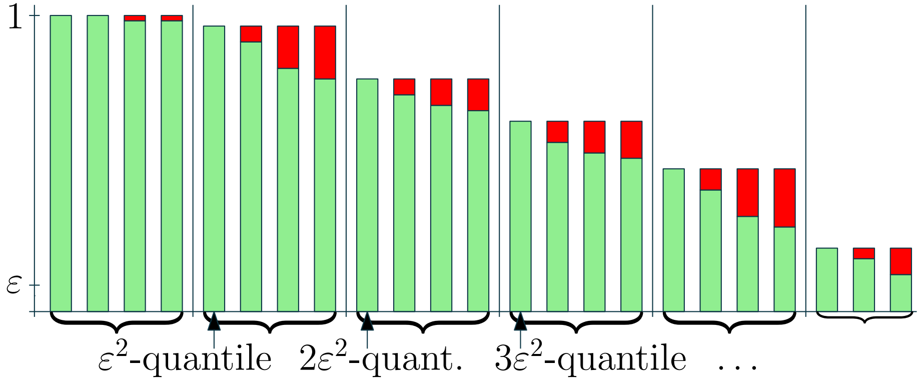
Big items: size $> \varepsilon$: **approximate linear grouping**



- Sufficient to δ-approximate quantiles $\delta \approx \varepsilon^2$
  - δ-approx. ϕ-quantile $= (\phi \pm \delta)N$-th largest item

Streaming $1+\varepsilon$-Approx. for Bin Packing

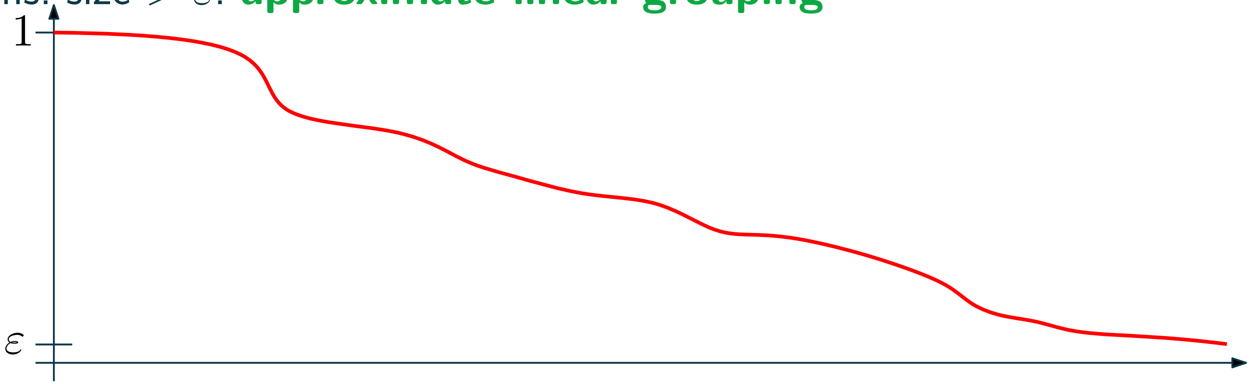
Big items: size $> \varepsilon$: **approximate linear grouping**



- Sufficient to δ-approximate quantiles $\delta \approx \varepsilon^2$
  - δ-approx. ϕ-quantile $= (\phi \pm \delta)N$-th largest item
- quantile summary w/ space $\mathcal{O}(\frac{1}{\delta} \cdot \log \delta N)$ [Greenwald & Khanna '01]
  - deterministic comparison-based

Streaming $1+\varepsilon$-Approx. for Bin Packing

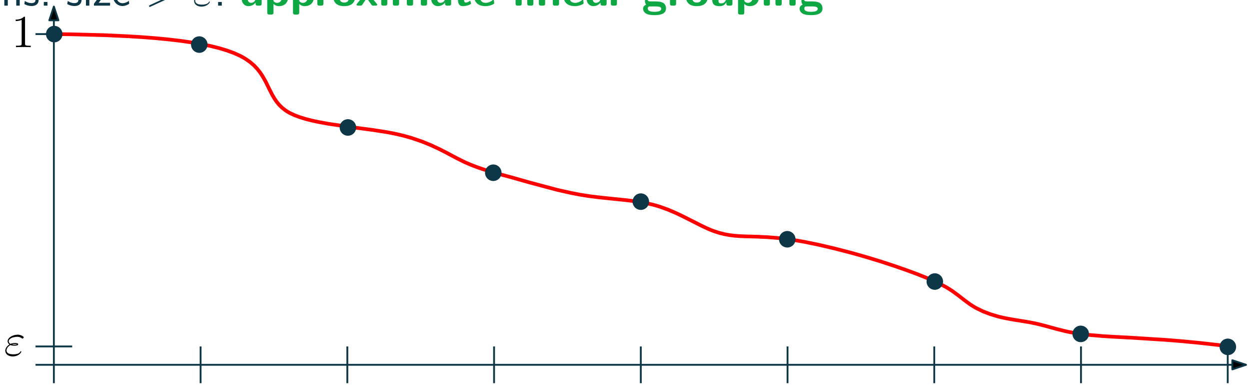
Big items: size $> \varepsilon$: **approximate linear grouping**



- Sufficient to δ-approximate quantiles $\delta \approx \varepsilon^2$
  - δ-approx. ϕ-quantile $= (\phi \pm \delta)N$-th largest item
- quantile summary w/ space $\mathcal{O}(\frac{1}{\delta} \cdot \log \delta N)$ [Greenwald & Khanna '01]
  - deterministic comparison-based

Streaming $1+\varepsilon$-Approx. for Bin Packing

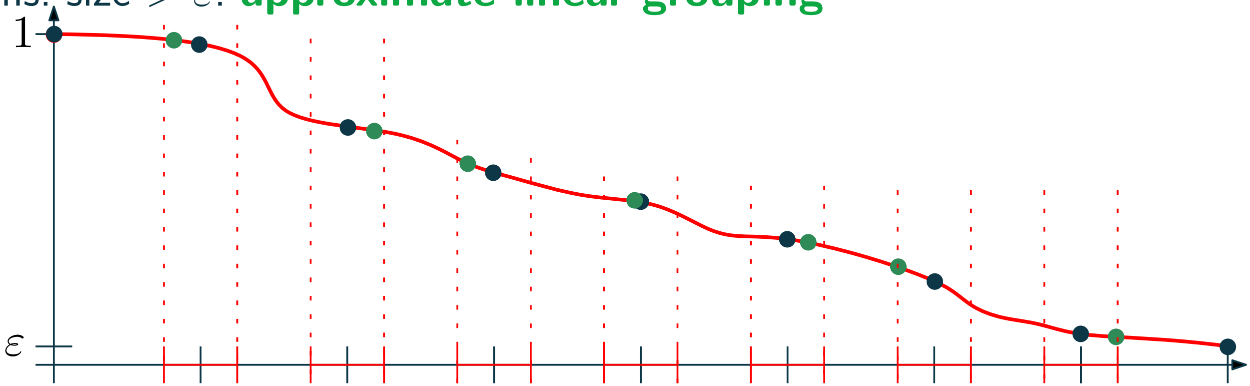
Big items: size $> \varepsilon$: **approximate linear grouping**



- Sufficient to δ-approximate quantiles $\delta \approx \varepsilon^2$
  - δ-approx. ϕ-quantile $= (\phi \pm \delta)N$-th largest item
- quantile summary w/ space $\mathcal{O}(\frac{1}{\delta} \cdot \log \delta N)$ [Greenwald & Khanna '01]
  - deterministic comparison-based

Streaming $1+\varepsilon$-Approx. for Bin Packing

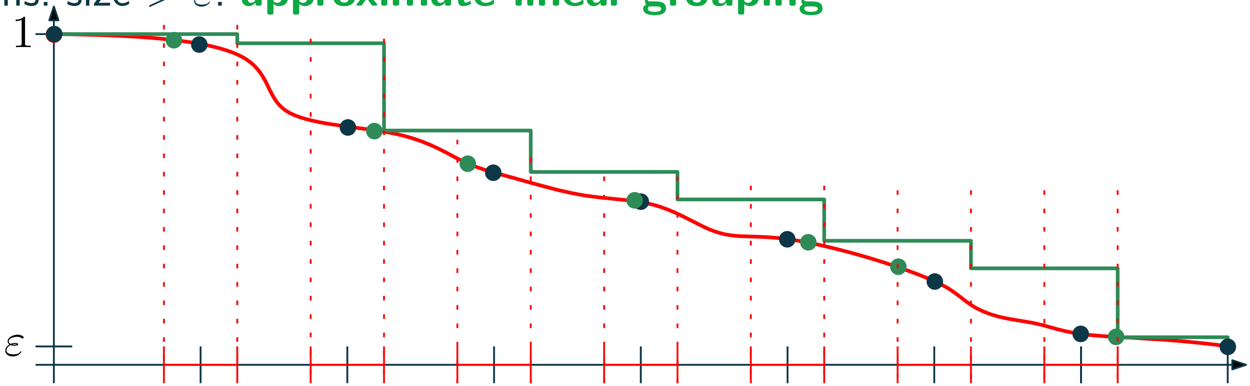
Big items: size $> \varepsilon$: **approximate linear grouping**



- Sufficient to δ-approximate quantiles $\delta \approx \varepsilon^2$
  - δ-approx. ϕ-quantile $= (\phi \pm \delta)N$-th largest item
- quantile summary w/ space $\mathcal{O}(\frac{1}{\delta} \cdot \log \delta N)$ [Greenwald & Khanna '01]
  - deterministic comparison-based

Streaming $1+\varepsilon$-Approx. for Bin Packing

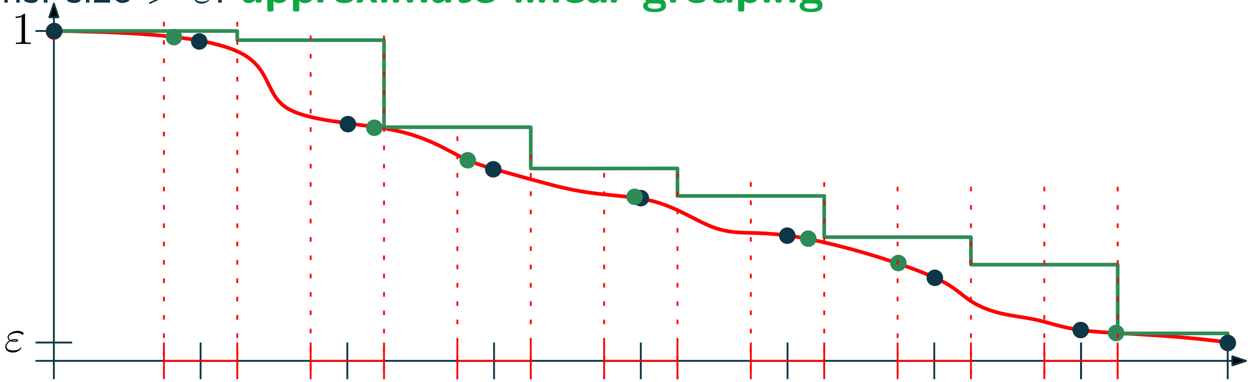
Big items: size $> \varepsilon$: **approximate linear grouping**



- Sufficient to δ-approximate quantiles $\delta \approx \varepsilon^2$
  - δ-approx. ϕ-quantile $= (\phi \pm \delta)N$-th largest item
- quantile summary w/ space $\mathcal{O}(\frac{1}{\delta} \cdot \log \delta N)$ [Greenwald & Khanna '01]
  - deterministic comparison-based

Streaming $1+\varepsilon$-Approx. for Bin Packing

Big items: size $> \varepsilon$: **approximate linear grouping**



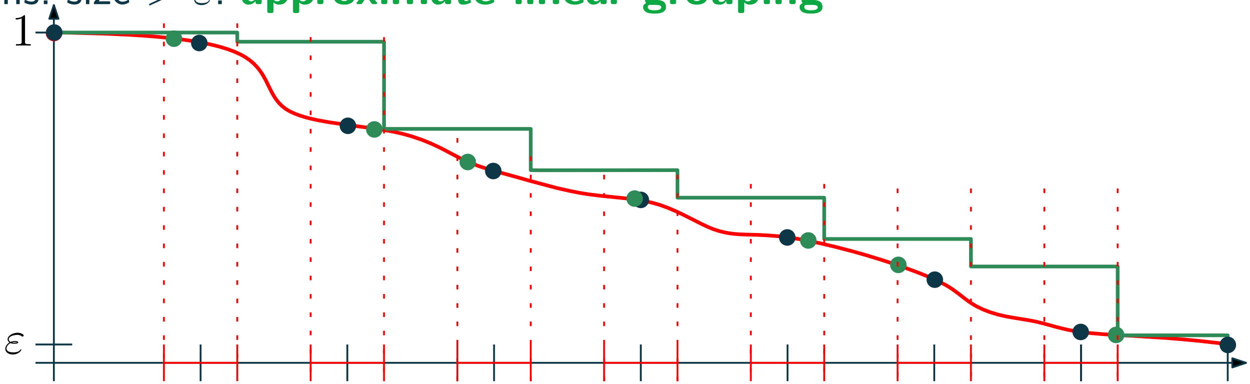
- Sufficient to δ-approximate quantiles $\delta \approx \varepsilon^2$
  - δ-approx. ϕ-quantile $= (\phi \pm \delta)N$-th largest item
- quantile summary w/ space $\mathcal{O}(\frac{1}{\delta} \cdot \log \delta N)$ [Greenwald & Khanna '01]
  - deterministic comparison-based

Small items

- sum their sizes
- fill in fractionally

Streaming $1+\varepsilon$-Approx. for Bin Packing

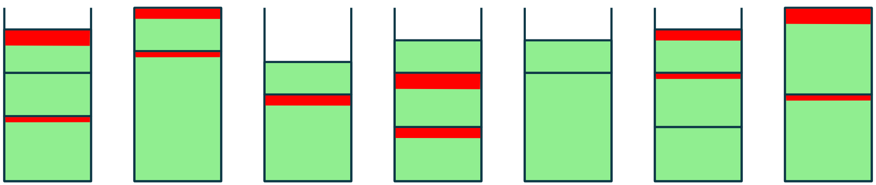
Big items: size $> \varepsilon$: **approximate linear grouping**



- Sufficient to δ-approximate quantiles $\delta \approx \varepsilon^2$
  - δ-approx. ϕ-quantile $= (\phi \pm \delta)N$-th largest item
- quantile summary w/ space $\mathcal{O}(\frac{1}{\delta} \cdot \log \delta N)$ [Greenwald & Khanna '01]
  - deterministic comparison-based

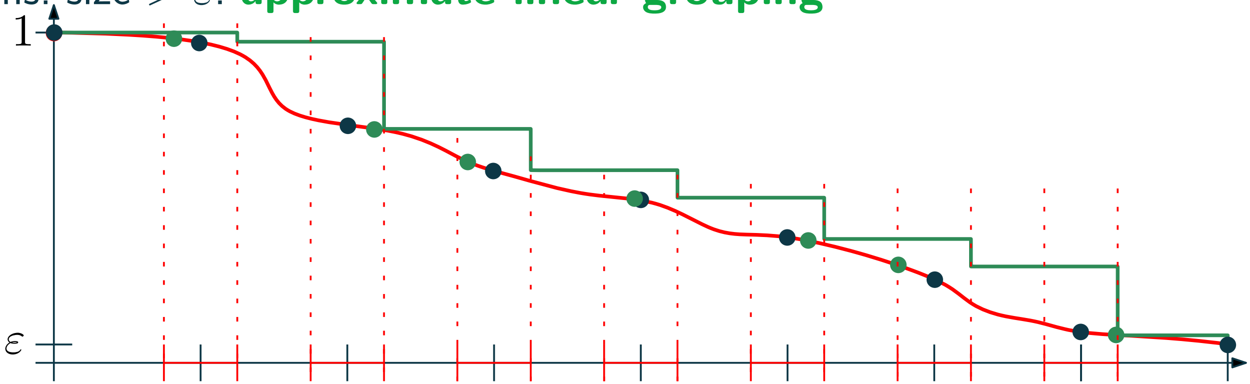
Small items

- sum their sizes
- fill in fractionally



Streaming $1 + \varepsilon$-Approx. for Bin Packing

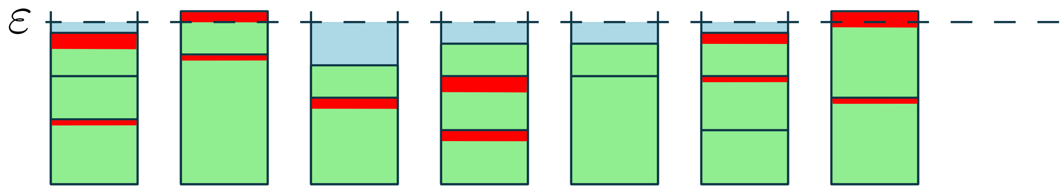
Big items: size $> \varepsilon$: **approximate linear grouping**



- Sufficient to δ-approximate quantiles $\delta \approx \varepsilon^2$
  - δ-approx. ϕ-quantile $= (\phi \pm \delta)N$-th largest item
- quantile summary w/ space $\mathcal{O}(\frac{1}{\delta} \cdot \log \delta N)$ [Greenwald & Khanna '01]
  - deterministic comparison-based

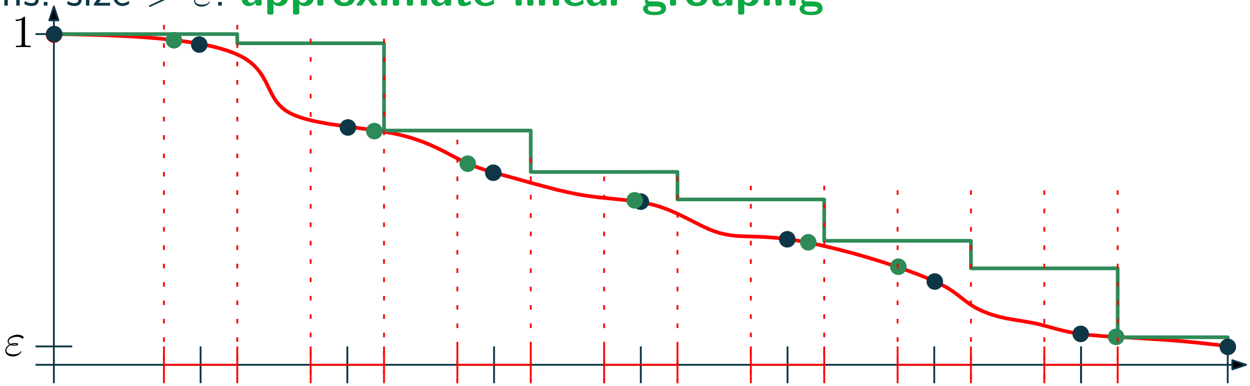
Small items

- sum their sizes
- fill in fractionally



Streaming $1 + \varepsilon$-Approx. for Bin Packing

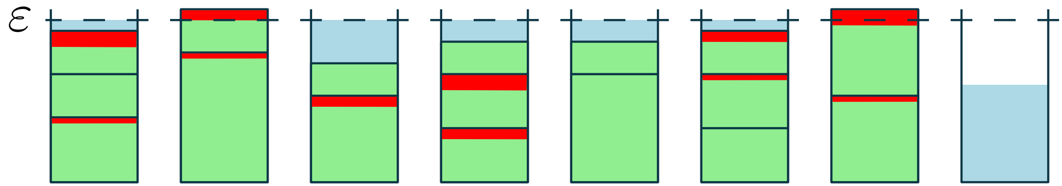
Big items: size $> \varepsilon$: **approximate linear grouping**



- Sufficient to δ-approximate quantiles $\delta \approx \varepsilon^2$
  - δ-approx. ϕ-quantile $= (\phi \pm \delta)N$-th largest item
- quantile summary w/ space $\mathcal{O}(\frac{1}{\delta} \cdot \log \delta N)$ [Greenwald & Khanna '01]
  - deterministic comparison-based

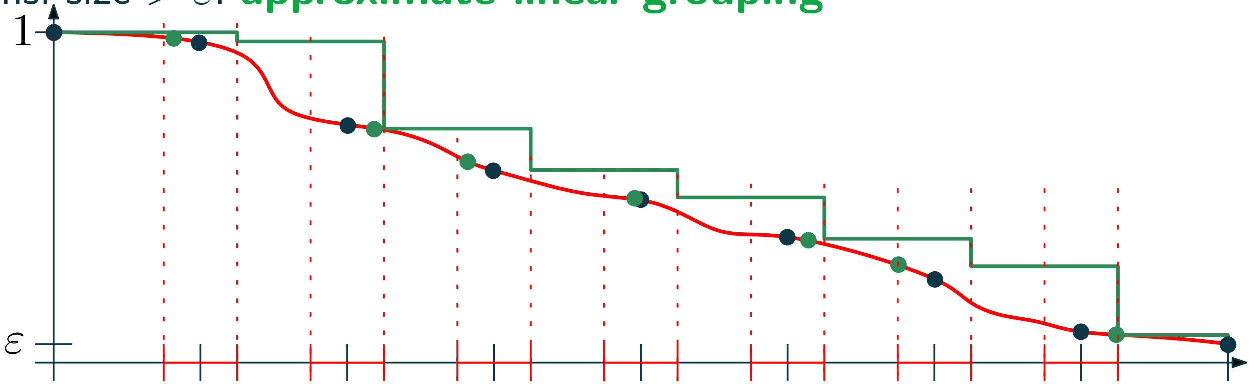
Small items

- sum their sizes
- fill in fractionally



Streaming $1+\varepsilon$-Approx. for Bin Packing

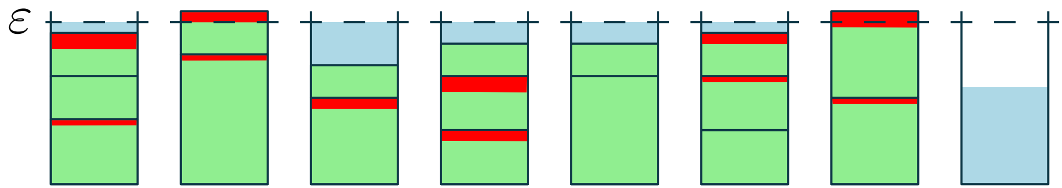
Big items: size $> \varepsilon$: **approximate linear grouping**



- Sufficient to δ-approximate quantiles $\delta \approx \varepsilon^2$
  - δ-approx. ϕ-quantile $= (\phi \pm \delta)N$-th largest item
- quantile summary w/ space $\mathcal{O}(\frac{1}{\delta} \cdot \log \delta N)$ [Greenwald & Khanna '01]
  - deterministic comparison-based

Small items

- sum their sizes
- fill in fractionally



$\Rightarrow 1+\varepsilon$-approx. for Bin Packing in space $\mathcal{O}(\frac{1}{\varepsilon^2} \cdot \log \varepsilon \mathrm{OPT})$

Streaming $1 + \varepsilon$-Approx. for Bin Packing

Streaming $1+\varepsilon$-Approx. for Bin Packing

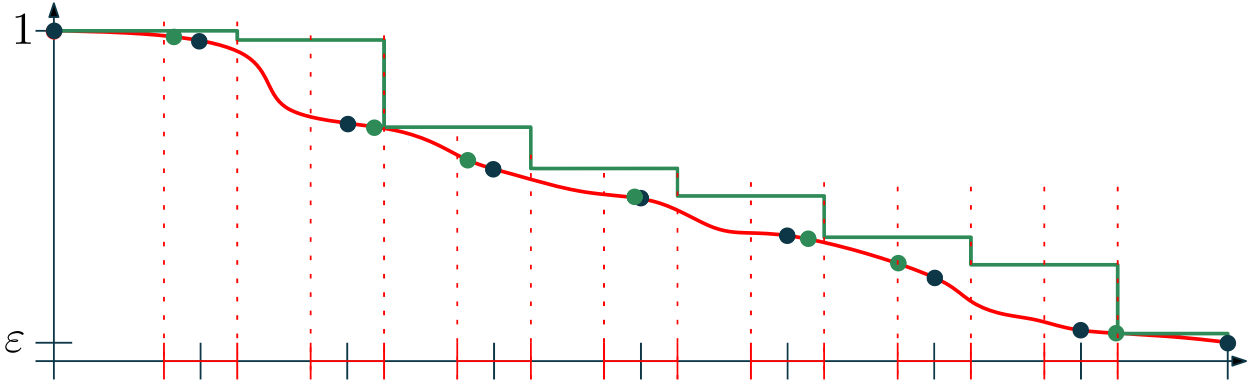
$1+\varepsilon$-approx. for $\textsc{Bin Packing}$ in space $\mathcal{O}(\frac{1}{\varepsilon^2} \cdot \log \varepsilon \mathsf{OPT})$

- by quantiles with precision $\approx \varepsilon^2$

Streaming $1 + \varepsilon$-Approx. for Bin Packing

$1 + \varepsilon$-approx. for $\textsc{Bin Packing}$ in space $\mathcal{O}(\frac{1}{\varepsilon^2} \cdot \log \varepsilon \mathrm{OPT})$

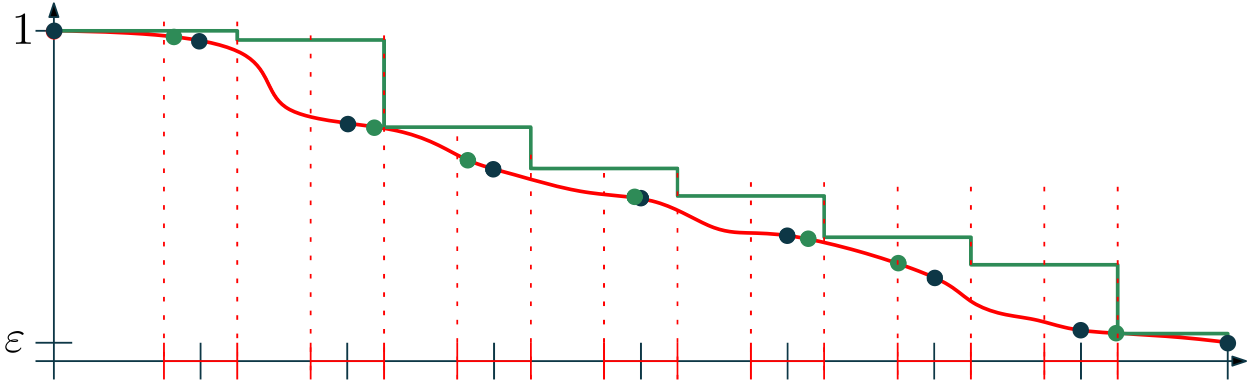
- by quantiles with precision $\approx \varepsilon^2$



Streaming $1 + \varepsilon$-Approx. for Bin Packing

$1 + \varepsilon$-approx. for Bin Packing in space $\mathcal{O}(\frac{1}{\varepsilon^2} \cdot \log \varepsilon \text{OPT})$

- by quantiles with precision $\approx \varepsilon^2$

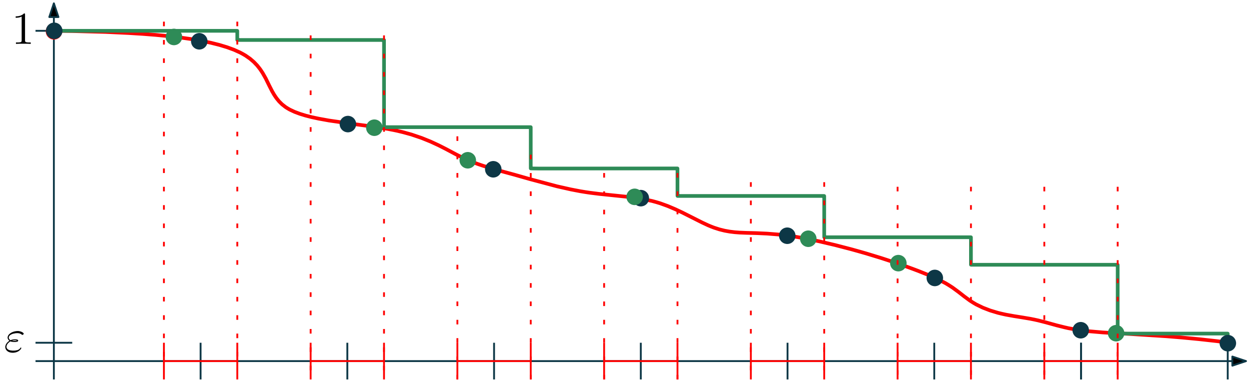


- too much precision for small items

Streaming $1 + \varepsilon$-Approx. for Bin Packing

$1 + \varepsilon$-approx. for Bin Packing in space $\mathcal{O}(\frac{1}{\varepsilon^2} \cdot \log \varepsilon \mathrm{OPT})$

- by quantiles with precision $\approx \varepsilon^2$



- too much precision for small items

Geometric grouping [Karmarkar, Karp '82]

- Split big items into $\lceil \log_2 \frac{1}{\varepsilon} \rceil$ size groups: $(\frac{1}{2}, 1], (\frac{1}{4}, \frac{1}{2}], \ldots$
- Use quantile summary for each group with precision $\approx \varepsilon$
- $\Rightarrow$ space $\mathcal{O}(\frac{1}{\varepsilon} \cdot \log \frac{1}{\varepsilon} \cdot \log \mathrm{OPT})$

Streaming $1+\varepsilon$-Approx. for Bin Packing

Can we do better than $\mathcal{O}(\frac{1}{\varepsilon} \cdot \log \frac{1}{\varepsilon} \cdot \log \mathrm{OPT})$?

Streaming $1+\varepsilon$-Approx. for Bin Packing

Can we do better than $\mathcal{O}(\frac{1}{\varepsilon} \cdot \log \frac{1}{\varepsilon} \cdot \log \mathrm{OPT})$?

Yes! If . . .

- . . . items drawn from a bounded-size universe U
  - space $\mathcal{O}(\frac{1}{\varepsilon} \cdot \log \frac{1}{\varepsilon} \cdot \log |U|)$ using a quantile summary from [Shrivastava *et al.* '04]

Streaming $1+\varepsilon$-Approx. for Bin Packing

Can we do better than $\mathcal{O}(\frac{1}{\varepsilon} \cdot \log \frac{1}{\varepsilon} \cdot \log \mathrm{OPT})$?

Yes! If . . .

- . . . items drawn from a bounded-size universe U
  - space $\mathcal{O}(\frac{1}{\varepsilon} \cdot \log \frac{1}{\varepsilon} \cdot \log |U|)$ using a quantile summary from [Shrivastava *et al.* '04]
- . . . randomization allowed (wrong answer w/ probability γ)
  - space $\mathcal{O}(\frac{1}{\varepsilon} \cdot \log \frac{1}{\varepsilon} \cdot \log \log \frac{\log \frac{1}{\varepsilon}}{\gamma})$ using a quantile summary from [Karnin *et al.* '16]

Streaming $1 + \varepsilon$-Approx. for Bin Packing

Can we do better than $\mathcal{O}(\frac{1}{\varepsilon} \cdot \log \frac{1}{\varepsilon} \cdot \log \mathrm{OPT})$?

Yes! If …

- … items drawn from a bounded-size universe U
  - space $\mathcal{O}(\frac{1}{\varepsilon} \cdot \log \frac{1}{\varepsilon} \cdot \log |U|)$ using a quantile summary from [Shrivastava *et al.* '04]
- … randomization allowed (wrong answer w/ probability γ)
  - space $\mathcal{O}(\frac{1}{\varepsilon} \cdot \log \frac{1}{\varepsilon} \cdot \log \log \frac{\log \frac{1}{\varepsilon}}{\gamma})$ using a quantile summary from [Karnin *et al.* '16]

No, not much by a deterministic comparison-based algo.

- Streaming $1 + \varepsilon$-approx. for Bin Packing in space S

  $\Rightarrow$ estimating rank w/ accuracy $\approx \varepsilon$ in space S

Streaming $1+\varepsilon$-Approx. for Bin Packing

Can we do better than $\mathcal{O}(\frac{1}{\varepsilon} \cdot \log \frac{1}{\varepsilon} \cdot \log \mathrm{OPT})$?

Yes! If ...

- ... items drawn from a bounded-size universe U
  - space $\mathcal{O}(\frac{1}{\varepsilon} \cdot \log \frac{1}{\varepsilon} \cdot \log |U|)$ using a quantile summary from [Shrivastava *et al.* '04]
- ... randomization allowed (wrong answer w/ probability γ)
  - space $\mathcal{O}(\frac{1}{\varepsilon} \cdot \log \frac{1}{\varepsilon} \cdot \log \log \frac{\log \frac{1}{\varepsilon}}{\gamma})$ using a quantile summary from [Karnin *et al.* '16]

No, not much by a deterministic comparison-based algo.

- Streaming $1+\varepsilon$-approx. for Bin Packing in space S

  $\Rightarrow$ estimating rank w/ accuracy $\approx \varepsilon$ in space S
- LB $\Omega(\frac{1}{\varepsilon} \cdot \log \varepsilon N)$ for estimating rank / quantile summaries [Cormode & **V.** '19+]

  $\Rightarrow$ LB $\Omega(\frac{1}{\varepsilon} \cdot \log \mathrm{OPT})$ for Bin Packing

Vector Scheduling: Rounding

Vector Scheduling: Rounding

- Input: jobs characterized by *d*-dimensional vectors
- Goal: assign jobs to machines to minimize makespan

  = maximum load over all machines and dimensions
- $1 + \varepsilon$-approximation (more involved rounding & MIP) [Bansal *et al.* '16]

Vector Scheduling: Rounding

- Input: jobs characterized by d-dimensional vectors
- Goal: assign jobs to machines to minimize makespan

  $=$ maximum load over all machines and dimensions
- $1 + \varepsilon$-approximation (more involved rounding & MIP) [Bansal *et al.* '16]
- Rescaling property: scaling every vector by $\alpha \Rightarrow$ scaling OPT by α

Vector Scheduling: Rounding

- Input: jobs characterized by d-dimensional vectors
- Goal: assign jobs to machines to minimize makespan

  $=$ maximum load over all machines and dimensions
- $1 + \varepsilon$-approximation (more involved rounding & MIP) [Bansal *et al.* '16]
- Rescaling property: scaling every vector by $\alpha \Rightarrow$ scaling OPT by α

Makespan Scheduling: $d = 1$

- Rounding currently big jobs to powers of $1 + \varepsilon$
  - Big jobs $=$ bigger than ε times currently largest job

Vector Scheduling: Rounding

- Input: jobs characterized by d-dimensional vectors
- Goal: assign jobs to machines to minimize makespan

  $=$ maximum load over all machines and dimensions
- $1 + \varepsilon$-approximation (more involved rounding & MIP) [Bansal *et al.* '16]
- Rescaling property: scaling every vector by $\alpha \Rightarrow$ scaling OPT by α

Makespan Scheduling: $d = 1$

- Rounding currently big jobs to powers of $1 + \varepsilon$
  - Big jobs $=$ bigger than ε times currently largest job
  - $\Rightarrow$ streaming $1 + \varepsilon$-approx. in space $\approx \log_{1+\varepsilon} \frac{1}{\varepsilon} = \mathcal{O}(\frac{1}{\varepsilon} \cdot \log \frac{1}{\varepsilon})$

Vector Scheduling: Rounding

- Input: jobs characterized by d-dimensional vectors
- Goal: assign jobs to machines to minimize makespan

  $=$ maximum load over all machines and dimensions
- $1 + \varepsilon$-approximation (more involved rounding & MIP) [Bansal *et al.* '16]
- Rescaling property: scaling every vector by $\alpha \Rightarrow$ scaling OPT by α

Makespan Scheduling: $d = 1$

- Rounding currently big jobs to powers of $1 + \varepsilon$
  - Big jobs $=$ bigger than ε times currently largest job
  - $\Rightarrow$ streaming $1 + \varepsilon$-approx. in space $\approx \log_{1+\varepsilon} \frac{1}{\varepsilon} = \mathcal{O}(\frac{1}{\varepsilon} \cdot \log \frac{1}{\varepsilon})$

Vector Scheduling: $d > 1$

- More intricate rounding from [Bansal *et al.* '16]:
  - Round to 0 coordinates small relatively to $\|v\|_\infty$
  - Big jobs: round each dimension to power of $1 + \varepsilon \Rightarrow$ space $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})^d$
  - Small jobs: round relative to $\|v\|_\infty$

Vector Scheduling: Aggregation Algorithm

Job vector v big if $\|v\|_\infty > \gamma\cdot$ (LB on OPT) for $\gamma \approx \varepsilon^2 / \log \frac{d}{\varepsilon}$

- $\Rightarrow$ at most $d \cdot m/\gamma$ big vectors
- Store them all

Vector Scheduling: Aggregation Algorithm

Job vector v big if $\|v\|_\infty > \gamma\cdot$ (LB on OPT) for $\gamma \approx \varepsilon^2/\log\frac{d}{\varepsilon}$

- $\Rightarrow$ at most $d \cdot m/\gamma$ big vectors
- Store them all

Small vectors:

- combine into containers $\approx$ big vectors

Vector Scheduling: Aggregation Algorithm

Job vector v big if $\|v\|_\infty > \gamma \cdot$ (LB on OPT) for $\gamma \approx \varepsilon^2 / \log \frac{d}{\varepsilon}$

- $\Rightarrow$ at most $d \cdot m/\gamma$ big vectors
- Store them all

Small vectors:

- combine into containers $\approx$ big vectors
- Maintain one open container
  - add incoming small vectors into it
  - close it once it becomes full & open a new one

Vector Scheduling: Aggregation Algorithm

Job vector v big if $\|v\|_\infty > \gamma \cdot$ (LB on OPT) for $\gamma \approx \varepsilon^2 / \log \frac{d}{\varepsilon}$

- $\Rightarrow$ at most $d \cdot m/\gamma$ big vectors
- Store them all

Small vectors:

- combine into containers $\approx$ big vectors
- Maintain one open container
  - add incoming small vectors into it
  - close it once it becomes full & open a new one

Analysis:

- Use the best schedule for big vectors: makespan $\leq$ OPT

Vector Scheduling: Aggregation Algorithm

Job vector v big if $\|v\|_\infty > \gamma\cdot$ (LB on OPT) for $\gamma \approx \varepsilon^2/\log\frac{d}{\varepsilon}$

- $\Rightarrow$ at most $d \cdot m/\gamma$ big vectors
- Store them all

Small vectors:

- combine into containers $\approx$ big vectors
- Maintain one open container
  - add incoming small vectors into it
  - close it once it becomes full & open a new one

Analysis:

- Use the best schedule for big vectors: makespan $\leq$ OPT
- Assign containers by an optimal online algorithm from [Im *et al.* '15]
  - Randomized & greedy assignment
  - Containers small $\Rightarrow$ nearly balanced assignment $\Rightarrow$ makespan $\leq (1+\varepsilon)\cdot$ OPT

Vector Scheduling: Aggregation Algorithm

Job vector v big if $\|v\|_\infty > \gamma\cdot$ (LB on OPT) for $\gamma \approx \varepsilon^2/\log\frac{d}{\varepsilon}$

- $\Rightarrow$ at most $d \cdot m/\gamma$ big vectors
- Store them all

Small vectors:

- combine into containers $\approx$ big vectors
- Maintain one open container
  - add incoming small vectors into it
  - close it once it becomes full & open a new one

Analysis:

- Use the best schedule for big vectors: makespan $\leq$ OPT
- Assign containers by an optimal online algorithm from [Im *et al.* '15]
  - Randomized & greedy assignment
  - Containers small $\Rightarrow$ nearly balanced assignment $\Rightarrow$ makespan $\leq (1+\varepsilon)\cdot$ OPT
- Combine the two cases: makespan $\leq (2 - \frac{1}{m} + \varepsilon)\cdot$ OPT

Vector Scheduling: Aggregation Algorithm

Job vector v big if $\|v\|_\infty > \gamma \cdot$ (LB on OPT) for $\gamma \approx \varepsilon^2 / \log \frac{d}{\varepsilon}$

- $\Rightarrow$ at most $d \cdot m/\gamma$ big vectors
- Store them all

Small vectors:

- combine into containers $\approx$ big vectors
- Maintain one open container
  - add incoming small vectors into it
  - close it once it becomes full & open a new one

Analysis:

- Use the best schedule for big vectors: makespan $\leq$ OPT
- Assign containers by an optimal online algorithm from [Im *et al.* '15]
  - Randomized & greedy assignment
  - Containers small $\Rightarrow$ nearly balanced assignment $\Rightarrow$ makespan $\leq (1+\varepsilon) \cdot$ OPT
- Combine the two cases: makespan $\leq (2 - \frac{1}{m} + \varepsilon) \cdot$ OPT

tight!

Conclusions & Open Problems

Bin Packing

- Streaming $1 + \varepsilon$-approximation in space $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$
- Tight up to $\mathcal{O}(\log \frac{1}{\varepsilon})$ factor by connection to quantile summaries

Conclusions & Open Problems

Bin Packing

- Streaming $1+\varepsilon$-approximation in space $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$
- Tight up to $\mathcal{O}(\log\frac{1}{\varepsilon})$ factor by connection to quantile summaries
- Streaming $d+\varepsilon$-approx. for $\textsc{Vector Bin Packing}$ in space $\widetilde{\mathcal{O}}(\frac{d}{\varepsilon})$

Conclusions & Open Problems

Bin Packing

- Streaming $1+\varepsilon$-approximation in space $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$
- Tight up to $\mathcal{O}(\log \frac{1}{\varepsilon})$ factor by connection to quantile summaries
- Streaming $d+\varepsilon$-approx. for Vector Bin Packing in space $\widetilde{\mathcal{O}}(\frac{d}{\varepsilon})$

Vector Scheduling

- Rounding gives $1+\varepsilon$-approximation in space exceeding $\left(\frac{1}{\varepsilon}\right)^d$ [Bansal *et al.* '16]

Conclusions & Open Problems

Bin Packing

- Streaming $1+\varepsilon$-approximation in space $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$
- Tight up to $\mathcal{O}(\log \frac{1}{\varepsilon})$ factor by connection to quantile summaries
- Streaming $d+\varepsilon$-approx. for Vector Bin Packing in space $\widetilde{\mathcal{O}}(\frac{d}{\varepsilon})$

Vector Scheduling

- Rounding gives $1+\varepsilon$-approximation in space exceeding $\left(\frac{1}{\varepsilon}\right)^d$ [Bansal *et al.* '16]
- Aggregation gives $2-\frac{1}{m}+\varepsilon$-approximation in space $\mathcal{O}(\frac{1}{\varepsilon^2}\cdot d^2\cdot m\cdot \log\frac{d}{\varepsilon})$

Conclusions & Open Problems

Bin Packing

- Streaming $1+\varepsilon$-approximation in space $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$
- Tight up to $\mathcal{O}(\log \frac{1}{\varepsilon})$ factor by connection to quantile summaries
- Streaming $d+\varepsilon$-approx. for Vector Bin Packing in space $\widetilde{\mathcal{O}}(\frac{d}{\varepsilon})$

Vector Scheduling

- Rounding gives $1+\varepsilon$-approximation in space exceeding $\left(\frac{1}{\varepsilon}\right)^d$ [Bansal *et al.* '16]
- Aggregation gives $2-\frac{1}{m}+\varepsilon$-approximation in space $\mathcal{O}(\frac{1}{\varepsilon^2}\cdot d^2\cdot m\cdot\log\frac{d}{\varepsilon})$

Better analysis of aggregation algorithm?

$O(1)$-approx. in space poly(d)?

Conclusions & Open Problems

Bin Packing

- Streaming $1 + \varepsilon$-approximation in space $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$
- Tight up to $\mathcal{O}(\log \frac{1}{\varepsilon})$ factor by connection to quantile summaries
- Streaming $d + \varepsilon$-approx. for Vector Bin Packing in space $\widetilde{\mathcal{O}}(\frac{d}{\varepsilon})$

Vector Scheduling

- Rounding gives $1 + \varepsilon$-approximation in space exceeding $\left(\frac{1}{\varepsilon}\right)^d$ [Bansal *et al.* '16]
- Aggregation gives $2 - \frac{1}{m} + \varepsilon$-approximation in space $\mathcal{O}(\frac{1}{\varepsilon^2} \cdot d^2 \cdot m \cdot \log \frac{d}{\varepsilon})$

Better analysis of aggregation algorithm?

$O(1)$-approx. in space poly(d)?

What about your favourite problem?

Conclusions & Open Problems

Bin Packing

- Streaming $1+\varepsilon$-approximation in space $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$
- Tight up to $\mathcal{O}(\log\frac{1}{\varepsilon})$ factor by connection to quantile summaries
- Streaming $d+\varepsilon$-approx. for Vector Bin Packing in space $\widetilde{\mathcal{O}}(\frac{d}{\varepsilon})$

Vector Scheduling

- Rounding gives $1+\varepsilon$-approximation in space exceeding $\left(\frac{1}{\varepsilon}\right)^d$ [Bansal *et al.* '16]
- Aggregation gives $2-\frac{1}{m}+\varepsilon$-approximation in space $\mathcal{O}(\frac{1}{\varepsilon^2}\cdot d^2\cdot m\cdot\log\frac{d}{\varepsilon})$

Better analysis of aggregation algorithm?

$O(1)$-approx. in space poly(d)?

What about your favourite problem?

Thank You!

Streaming vs. Online

Streaming vs. Online

Problem	Streaming apx.	Competitive ratio LB	Competitive ratio UB
Bin Packing	$1+\varepsilon$	1.542 [Balogh *et al.* '19]	1.578 [Balogh *et al.* '18]
Vector Bin Packing	$d+\varepsilon$	$\Omega(d^{1-\varepsilon})$ [Azar *et al.* '13]	$d+0.7$ [Garey *et al.* '76]
Makespan Scheduling	$1+\varepsilon$	1.88 [Rudin '01]	1.92 [Fleischer & Wahl '00]
Vector Scheduling	$2\,(1+\varepsilon)$	$\Omega(\log d/\log\log d)$ [Im *et al.* '15]	$\mathcal{O}(\log d/\log\log d)$ [Im *et al.* '15]