A Faster Exponential Time Algorithm for Bin Packing with Constant Number of Bins

To appear at SODA 2021

Jesper Nederlof

UU

Jakub Pawlewicz

Warsaw

Céline Swennenhuis

TU/e

Karol Węgrzycki

Warsaw/Saarbrücken

Started with this research at "Parameterized Algorithms Retreat of the University of Warsaw" in February 2020

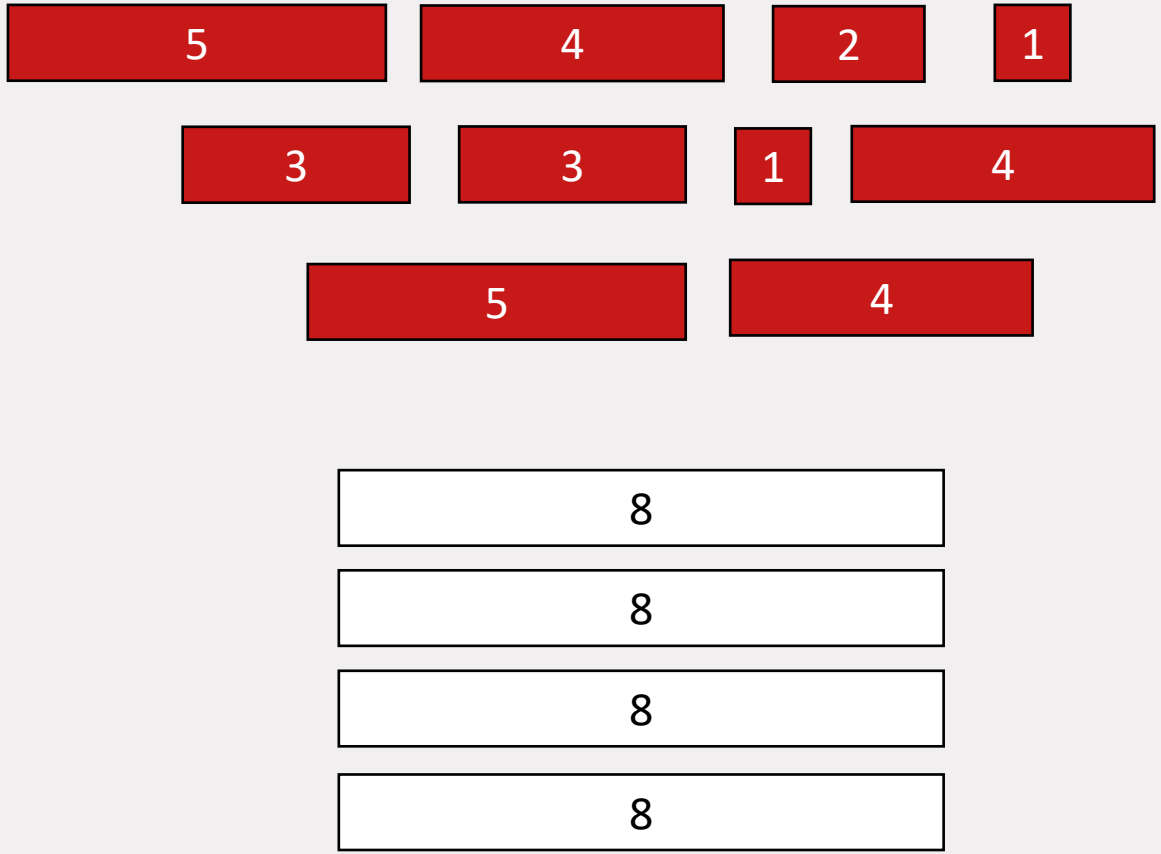
Department of Mathematics and Computer Science, Eindhoven University of Technology

Bin Packing

Given:

- n items
- $w(j)$ weight of item j
  - $w(X) = \sum_{j \in X} w(j)$
- m bins with capacity c

Goal: **distribute items over bins**



Bin Packing

- Dynamic Programming
  - $O(c^m \cdot n)$

- Björklund, Husfeldt and Koivisto (SICOMP 2009)
  - $O(2^n \cdot n)$

Open Question

Can do in $O(1.99999^n)$ time?

Before our work, only known for $m = 2,3$ (Lente et al.)

Key idea:

DP in $O\left(2^{(1-\varepsilon_m)n}\right)$ time

or

Björklund et al. in $O\left(2^{(1-\varepsilon_m)n}\right)$ time

Our result:

Can do in $O\left(2^{(1-\varepsilon_m)n}\right)$ time with $\varepsilon_m > 0$ that depends on m.

Dynamic Program

$$T(j, c_1, c_2, \dots, c_m) = \bigvee_{i \in [m]} T(j-1, \dots, c_i - w(j), \dots)$$

Items $1, \dots, j$

leftover capacities

$O(c^m \cdot n)$

$\mathcal{W} := \{w(X) : X \subseteq [n]\}$

Runtime: $O(|\mathcal{W}|^m \cdot n)$

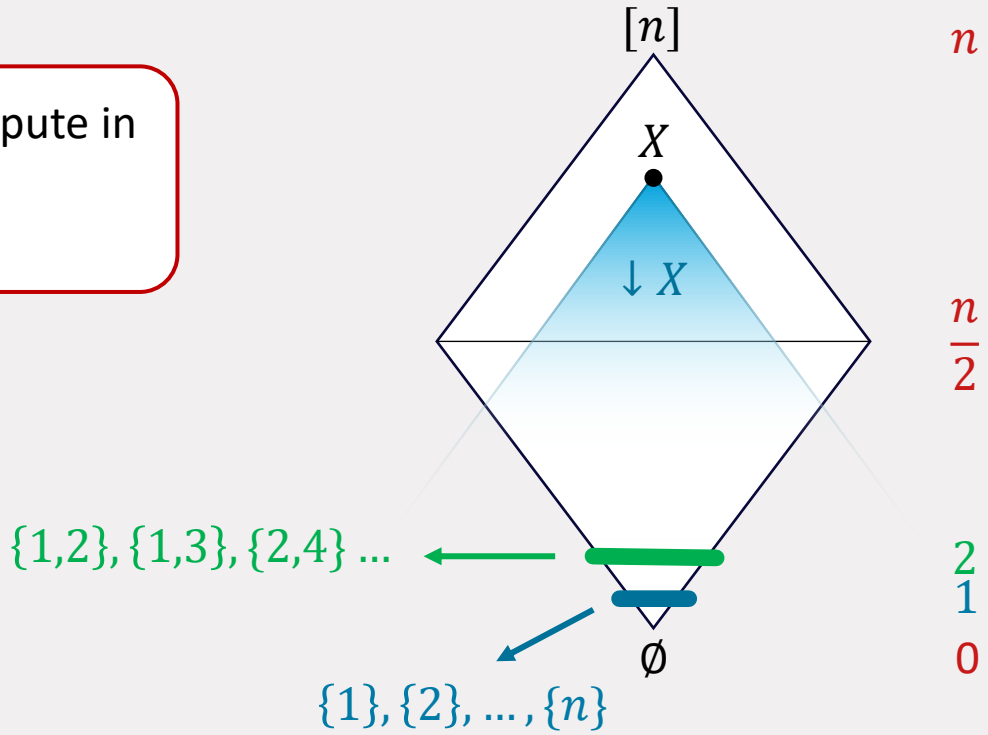
$$\{2,2,2,\dots,2\} \rightarrow \mathcal{W} = \{2,4,\dots,2n\}$$

$$\{1,2,\dots,2^{n-1}\} \rightarrow \mathcal{W} = \{1,2,3,\dots,2^n - 1\}$$

If $|\mathcal{W}| \leq 2^{\frac{(1-\gamma)n}{m}}$, runtime $O(2^{(1-\gamma)n})$!

Algorithm by Björklund et al.

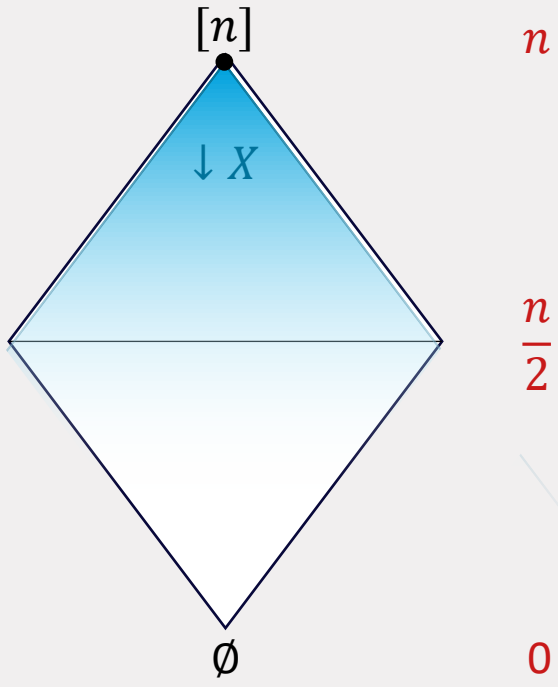
Theorem: For $X \subseteq [n]$, we can compute in time $O(|\downarrow X| \cdot n)$ whether X can be distributed over m bins.



Algorithm by Björklund et al.

Theorem: For $X \subseteq [n]$, we can compute in time $O(|\downarrow X| \cdot n)$ whether X can be distributed over m bins.

If $X = [n]$, then in time $O(2^n \cdot n)$!



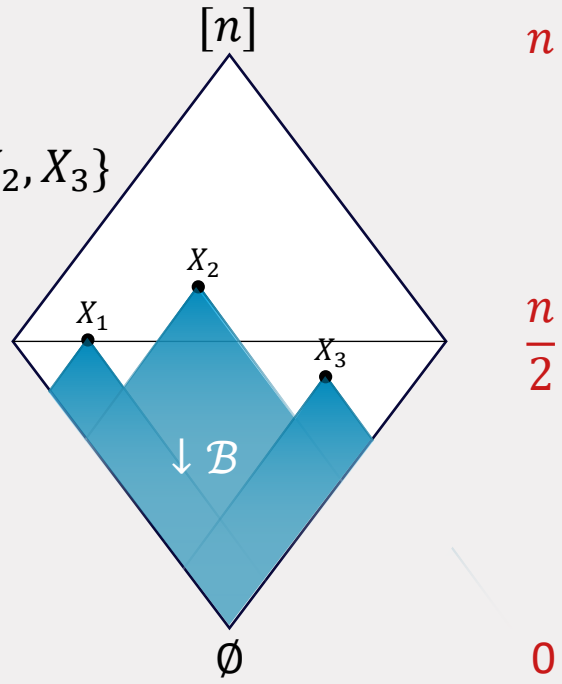
Algorithm by Björklund et al.

Theorem: For $X \subseteq [n]$, we can compute in time $O(|\downarrow X| \cdot n)$ whether X can be distributed over m bins.

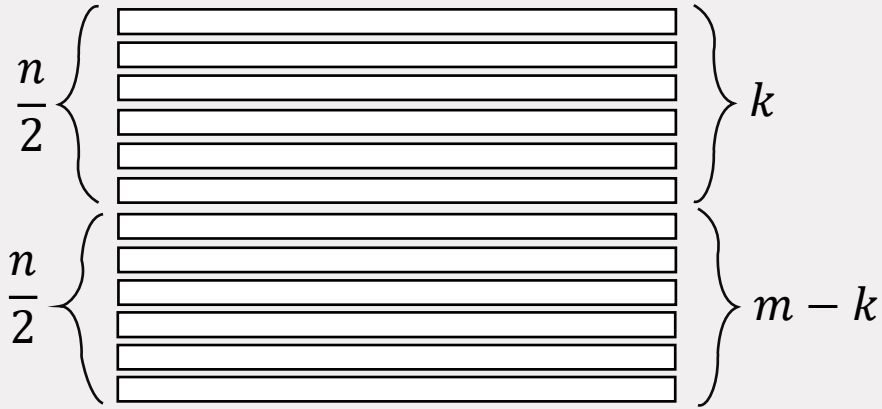
If $X = [n]$, then in time $O(2^n \cdot n)$!

Theorem: For $\boldsymbol{\mathcal{B}} \subseteq \mathbf{2}^{\boldsymbol{n}}$, we can compute in time $O(|\downarrow \boldsymbol{\mathcal{B}}| \cdot n)$, whether X can be distributed over m bins **for all $\boldsymbol{X} \in \boldsymbol{\mathcal{B}}$**.

$\mathcal{B} = \{X_1, X_2, X_3\}$

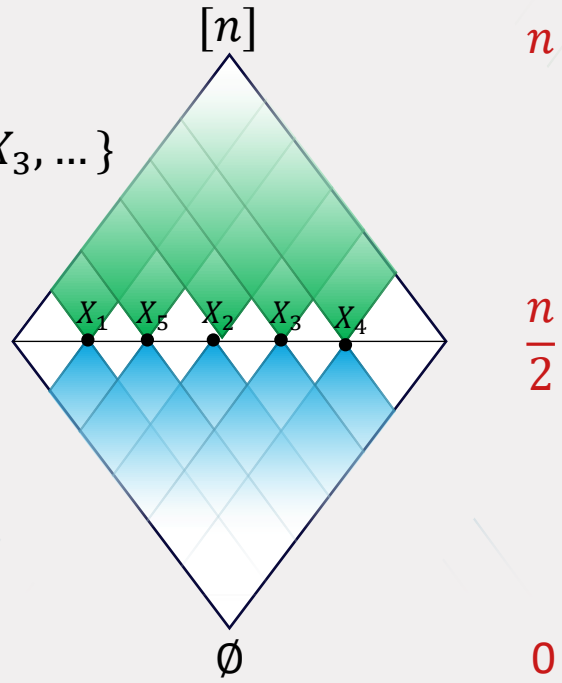


Algorithm by Björklund et al.



$$\mathcal{B} := \left\{X : |X| = \frac{n}{2}, w(X) = k \cdot c\right\}$$

$$\mathcal{B} = \{X_1, X_2, X_3, \dots\}$$



Compute in time $O(|\downarrow \mathcal{B}| \cdot n)$ which X can be put in k bins

Compute in time $O(|\uparrow \mathcal{B}| \cdot n)$ which $[n] \setminus X$ can be put in $m-k$ bins

Algorithm by Björklund et al.

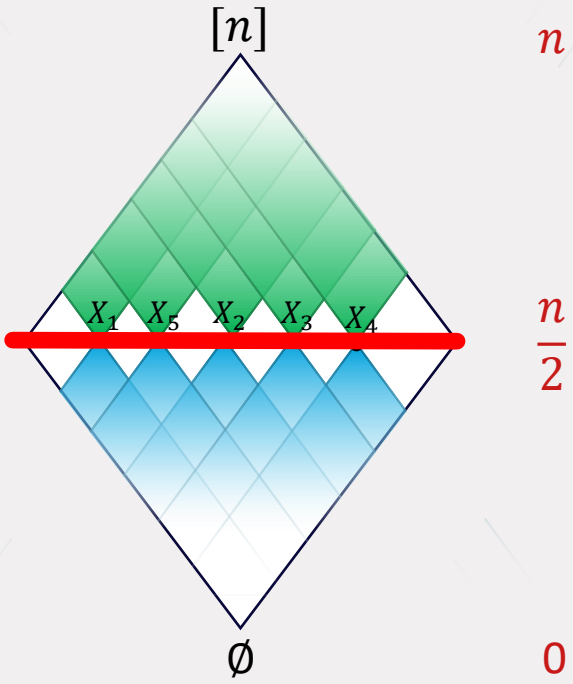
$$\{2,2,2,\dots,2\}$$

$$\mathcal{B} = \left|\left\{X : |X| = \frac{n}{2}, \omega(X) = n\right\}\right| = \binom{n}{n/2}$$

$$\{1,2,4,\dots,2^{n-1}\}$$

$$\mathcal{B} \leq 1$$

$$\mathcal{B} := \left\{X : |X| = \frac{n}{2}, w(X) = k \cdot c\right\}$$



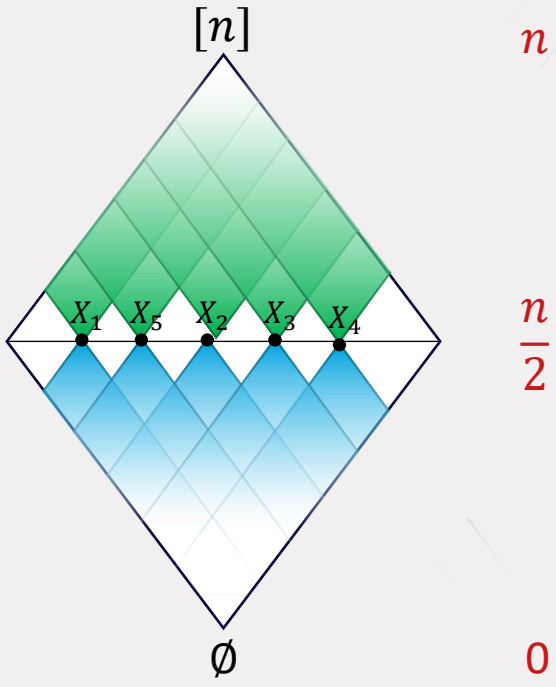
Algorithm by Björklund et al.

If	$\lvert\mathcal{B}\rvert \leq 2^{(1-\varepsilon)n}$	for $\varepsilon > 0$
Then	$\lvert\downarrow \mathcal{B}\rvert + \lvert\uparrow \mathcal{B}\rvert \leq 2^{(1-\delta_\varepsilon)n}$	for $\delta_\varepsilon > 0$

So if $\lvert\mathcal{B}\rvert \leq 2^{(1-\varepsilon)n}$,

we can do faster than 2^n!

$$\mathcal{B} := \left\{X : |X| = \frac{n}{2}, w(X) = k \cdot c\right\}$$



Additive Combinatorics

If $|\mathcal{W}| \leq 2^{\frac{(1-\gamma)n}{m}}$,
then DP fast

If $|\mathcal{B}| \leq 2^{(1-\varepsilon)n}$,
then Björklund et al. algorithm fast

Theorem: For all $\delta > 0$ there exists $\varepsilon > 0$ s.t.

if $|\mathcal{W}| \geq 2^{\delta n}$, then $|\mathcal{B}| \leq 2^{(1-\varepsilon)n}$.

Take $\delta = \frac{(1-\gamma)}{m}$, then either

$|\mathcal{W}| \leq 2^{\delta n}$ or $|\mathcal{B}| \leq 2^{(1-\varepsilon)n}$ *(or both)*

Conclusion

Main result:

Bin Packing in $O\left(2^{(1-\varepsilon_m)n}\right)$ time with $\varepsilon_m > 0$ that depends on m.

Key idea:

Tradeoff between $|\mathcal{W}|$ and $|\mathcal{B}|$.

Thanks for your attention!