A parameterized approximation scheme for the 2D-Knapsack problem with wide items

Mathieu Mari

University of Warsaw and IDEAS-NCBR

Now at LIRMM, Montpellier

Timothé Picavet

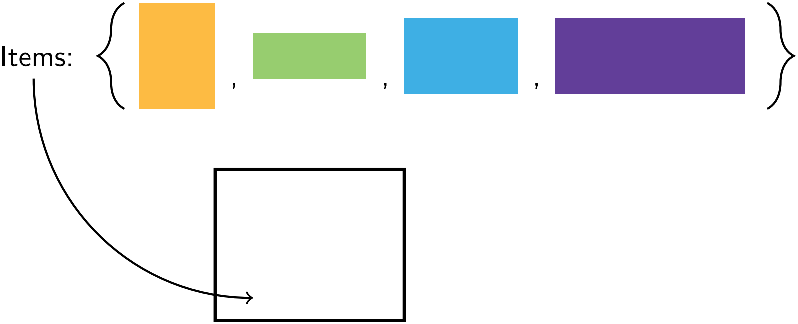
ENS de Lyon

Now at LaBRI, Bordeaux

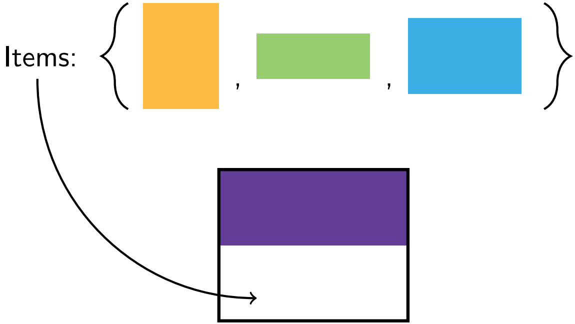
Michał Pilipczuk

University of Warsaw

What is 2D-KNAPSACK?

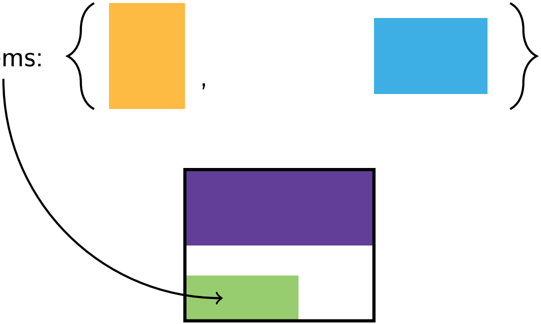


What is 2D-Knapsack?

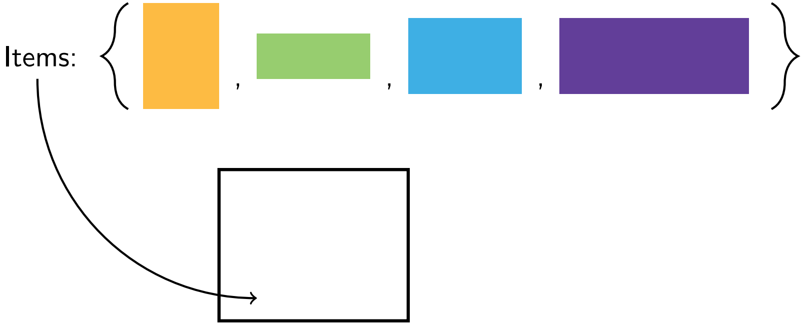


What is 2D-Knapsack?

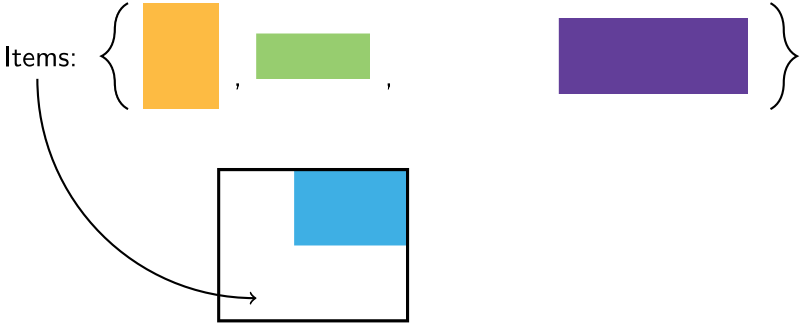
Items:



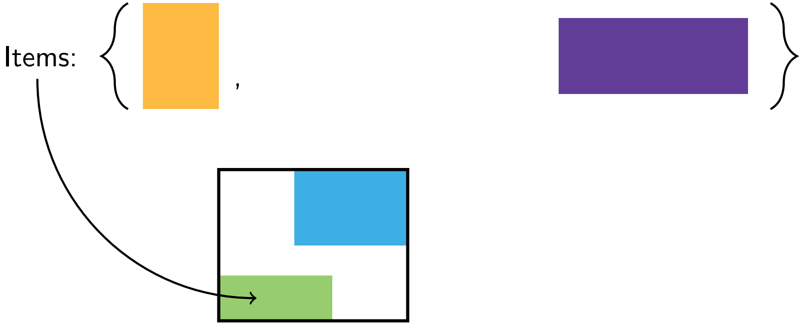
What is 2D-Knapsack?



What is 2D-KNAPSACK?



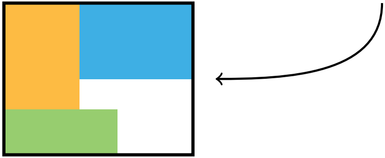
What is 2D-Knapsack?



What is 2D-KNAPSACK?

Items: 

✔ You've packed the maximum number of Items for Amsterdam!



What is known about 2D-Knapsack?

[1] F. Grandoni, S. Kratsch, A. Wiese. "Parameterized Approximation Schemes for Independent Set of Rectangles and Geometric Knapsack". ESA 2019

[2] A. Adamaszek, A. Wiese. "A quasi-PTAS for the Two-Dimensional Geometric Knapsack problem". SODA 2015

[3] W. Gálvez et al. "Improved Approximation Algorithms for 2-Dimensional Knapsack: Packing into Multiple L-Shapes, Spirals, and More". SoCG 2021

What is known about 2D-Knapsack?

- NP-Hard (by reduction to Knapsack)

[1] F. Grandoni, S. Kratsch, A. Wiese. “Parameterized Approximation Schemes for Independent Set of Rectangles and Geometric Knapsack”. ESA 2019

[2] A. Adamaszek, A. Wiese. “A quasi-PTAS for the Two-Dimensional Geometric Knapsack problem”. SODA 2015

[3] W. Gálvez et al. “Improved Approximation Algorithms for 2-Dimensional Knapsack: Packing into Multiple L-Shapes, Spirals, and More”. SoCG 2021

What is known about 2D-KNAPSACK?

- NP-Hard (by reduction to KNAPSACK)
- Even W[1]-hard parameterized by k[1]

[1] F. Grandoni, S. Kratsch, A. Wiese. "Parameterized Approximation Schemes for Independent Set of Rectangles and Geometric Knapsack". ESA 2019

[2] A. Adamaszek, A. Wiese. "A quasi-PTAS for the Two-Dimensional Geometric Knapsack problem". SODA 2015

[3] W. Gálvez et al. "Improved Approximation Algorithms for 2-Dimensional Knapsack: Packing into Multiple L-Shapes, Spirals, and More". SoCG 2021

What is known about 2D-Knapsack?

- NP-Hard (by reduction to Knapsack)
- Even W[1]-hard parameterized by k[1]
- Focus on the unary variant (w.r.t. dimensions of the box)

[1] F. Grandoni, S. Kratsch, A. Wiese. “Parameterized Approximation Schemes for Independent Set of Rectangles and Geometric Knapsack”. ESA 2019

[2] A. Adamaszek, A. Wiese. “A quasi-PTAS for the Two-Dimensional Geometric Knapsack problem”. SODA 2015

[3] W. Gálvez et al. “Improved Approximation Algorithms for 2-Dimensional Knapsack: Packing into Multiple L-Shapes, Spirals, and More”. SoCG 2021

What is known about 2D-Knapsack?

- NP-Hard (by reduction to Knapsack)
- Even W[1]-hard parameterized by k[1]
- Focus on the unary variant (w.r.t. dimensions of the box)
- In this setting:

[1] F. Grandoni, S. Kratsch, A. Wiese. "Parameterized Approximation Schemes for Independent Set of Rectangles and Geometric Knapsack". ESA 2019

[2] A. Adamaszek, A. Wiese. "A quasi-PTAS for the Two-Dimensional Geometric Knapsack problem". SODA 2015

[3] W. Gálvez et al. "Improved Approximation Algorithms for 2-Dimensional Knapsack: Packing into Multiple L-Shapes, Spirals, and More". SoCG 2021

What is known about 2D-KNAPSACK?

- NP-Hard (by reduction to KNAPSACK)
- Even W[1]-hard parameterized by k[1]
- Focus on the unary variant (w.r.t. dimensions of the box)
- In this setting:
  - QPTAS found[2]

[1] F. Grandoni, S. Kratsch, A. Wiese. "Parameterized Approximation Schemes for Independent Set of Rectangles and Geometric Knapsack". ESA 2019

[2] A. Adamaszek, A. Wiese. "A quasi-PTAS for the Two-Dimensional Geometric Knapsack problem". SODA 2015

[3] W. Gálvez et al. "Improved Approximation Algorithms for 2-Dimensional Knapsack: Packing into Multiple L-Shapes, Spirals, and More". SoCG 2021

What is known about 2D-Knapsack?

- NP-Hard (by reduction to Knapsack)
- Even W[1]-hard parameterized by k[1]
- Focus on the unary variant (w.r.t. dimensions of the box)
- In this setting:
  - QPTAS found[2]
  - Best approximation factor in polytime[3]: $4/3 + \varepsilon$

[1] F. Grandoni, S. Kratsch, A. Wiese. "Parameterized Approximation Schemes for Independent Set of Rectangles and Geometric Knapsack". ESA 2019

[2] A. Adamaszek, A. Wiese. "A quasi-PTAS for the Two-Dimensional Geometric Knapsack problem". SODA 2015

[3] W. Gálvez et al. "Improved Approximation Algorithms for 2-Dimensional Knapsack: Packing into Multiple L-Shapes, Spirals, and More". SoCG 2021

What is known about 2D-Knapsack?

- NP-Hard (by reduction to Knapsack)
- Even W[1]-hard parameterized by k[1]
- Focus on the unary variant (w.r.t. dimensions of the box)
- In this setting:
  - QPTAS found[2]
  - Best approximation factor in polytime[3]: $4/3 + \varepsilon$
- PTAS?

[1] F. Grandoni, S. Kratsch, A. Wiese. "Parameterized Approximation Schemes for Independent Set of Rectangles and Geometric Knapsack". ESA 2019

[2] A. Adamaszek, A. Wiese. "A quasi-PTAS for the Two-Dimensional Geometric Knapsack problem". SODA 2015

[3] W. Gálvez et al. "Improved Approximation Algorithms for 2-Dimensional Knapsack: Packing into Multiple L-Shapes, Spirals, and More". SoCG 2021

Parameterized Approximation Schemes

A parameterized approximation scheme (PAS):

[1] F. Grandoni, S. Kratsch, A. Wiese. "Parameterized Approximation Schemes for Independent Set of Rectangles and Geometric Knapsack". ESA 2019

Parameterized Approximation Schemes

A parameterized approximation scheme (PAS):

- finds a packing of size $\geq (1-\varepsilon)k$

[1] F. Grandoni, S. Kratsch, A. Wiese. "Parameterized Approximation Schemes for Independent Set of Rectangles and Geometric Knapsack". ESA 2019

Parameterized Approximation Schemes

A parameterized approximation scheme (PAS):

- finds a packing of size $\geq (1-\varepsilon)k$
- or correctly concludes $\nexists$ packing of size k,

[1] F. Grandoni, S. Kratsch, A. Wiese. "Parameterized Approximation Schemes for Independent Set of Rectangles and Geometric Knapsack". ESA 2019

Parameterized Approximation Schemes

A parameterized approximation scheme (PAS):

- finds a packing of size $\geq (1-\varepsilon)k$
- or correctly concludes $\nexists$ packing of size k,
- and runs in time $f_\varepsilon(k) \cdot n^{\mathcal{O}_\varepsilon(1)}$.

[1] F. Grandoni, S. Kratsch, A. Wiese. "Parameterized Approximation Schemes for Independent Set of Rectangles and Geometric Knapsack". ESA 2019

Parameterized Approximation Schemes

A parameterized approximation scheme (PAS):

- finds a packing of size $\geq (1-\varepsilon)k$
- or correctly concludes $\nexists$ packing of size k,
- and runs in time $f_\varepsilon(k) \cdot n^{\mathcal{O}_\varepsilon(1)}$.

Prior work: $k^{\mathcal{O}(k/\varepsilon)} \cdot n^{\mathcal{O}(1/\varepsilon^3)}$ when allowing 90° rotations[1]

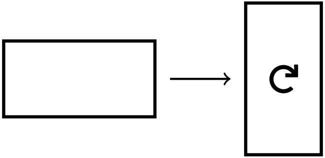
[1] F. Grandoni, S. Kratsch, A. Wiese. "Parameterized Approximation Schemes for Independent Set of Rectangles and Geometric Knapsack". ESA 2019

Parameterized Approximation Schemes

A parameterized approximation scheme (PAS):

- finds a packing of size $\geq (1-\varepsilon)k$
- or correctly concludes $\nexists$ packing of size k,
- and runs in time $f_\varepsilon(k) \cdot n^{\mathcal{O}_\varepsilon(1)}$.

Prior work: $k^{\mathcal{O}(k/\varepsilon)} \cdot n^{\mathcal{O}(1/\varepsilon^3)}$ when allowing 90° rotations[1]



[1] F. Grandoni, S. Kratsch, A. Wiese. "Parameterized Approximation Schemes for Independent Set of Rectangles and Geometric Knapsack". ESA 2019

Our Contribution

Theorem

2D-Knapsack *admits a PAS under the following assuptions:*

Our Contribution

Theorem

2D-Knapsack *admits a PAS under the following assuptions:*

- ≤ *Integral, polynomially bounded sidelengths (unary version)*

Our Contribution

Theorem

2D-Knapsack *admits a PAS under the following assuptions:*

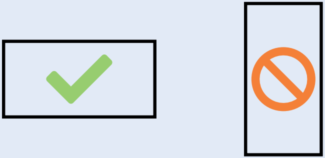
- ≤ *Integral, polynomially bounded sidelengths (unary version)*
- ↔ *Every rectangle is wide (width $\geq$ height)*

Our Contribution

Theorem

2D-KNAPSACK *admits a PAS under the following assuptions:*

- ≤ *Integral, polynomially bounded sidelengths (unary version)*
- ↔ *Every rectangle is wide (width $\geq$ height)*

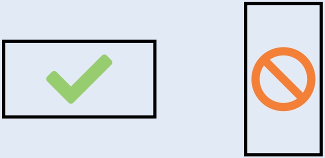


Our Contribution

Theorem

2D-KNAPSACK *admits a PAS under the following assuptions:*

- ≤ *Integral, polynomially bounded sidelengths (unary version)*
- ↔ *Every rectangle is wide (width $\geq$ height)*



- *Box aspect ratio is bounded by δ*

Our approach

1. Remove all small rectangles
   $\implies$ keep only rectangles of substantial width
   (width $\geq$ box width/$g(k, \varepsilon)$)

Our approach

1. ~~Remove all small rectangles~~
   ~~$\implies$ keep only rectangles of substantial width (width $\geq$ box width$/g(k,\varepsilon)$)~~

1. Wide assumption + remove 1 big rectangle
   $\implies$ keep only rectangles of substantial width (width $\geq$ box width$/g(k,\varepsilon)$)

Our approach

1. ~~Remove all small rectangles $\implies$ keep only rectangles of substantial width (width $\geq$ box width$/g(k, \varepsilon)$)~~
1. Wide assumption + remove 1 big rectangle $\implies$ keep only rectangles of substantial width (width $\geq$ box width$/g(k, \varepsilon)$)
2. **Substantial width + remove εk rectangles $\implies$ WLOG, solution structured**

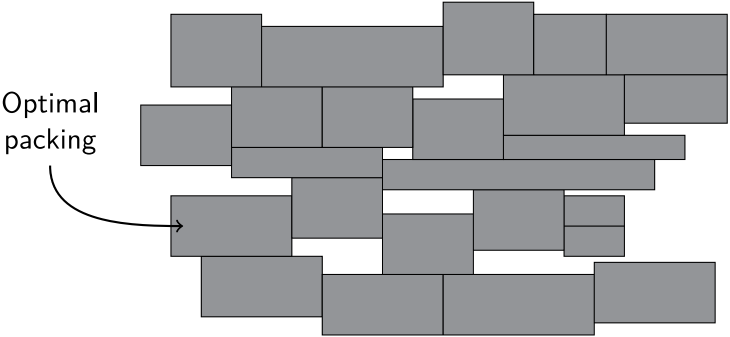
Our approach

1. ~~Remove all small rectangles $\implies$ keep only rectangles of substantial width (width $\geq$ box width$/g(k, \varepsilon)$)~~

1. Wide assumption + remove 1 big rectangle
   $\implies$ keep only rectangles of substantial width (width $\geq$ box width$/g(k, \varepsilon)$)
2. **Substantial width + remove εk rectangles**
   $\implies$ **WLOG, solution structured**
3. Dynamic Programming + color coding
   $\implies$ find the structured solution

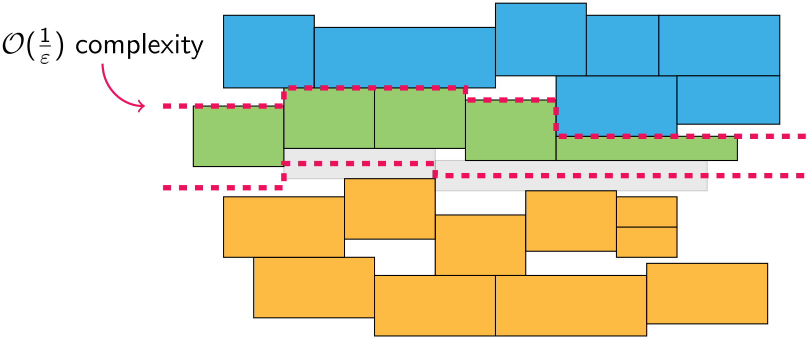
The structural lemma

If all rectangles have substantial width, by deleting εk rectangles:



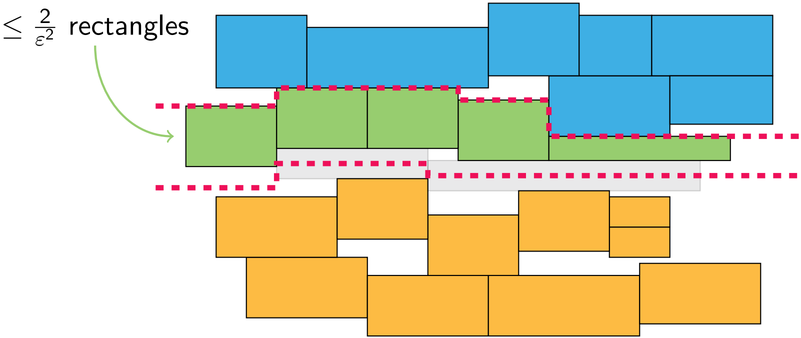
The structural lemma

If all rectangles have substantial width, by deleting εk rectangles:



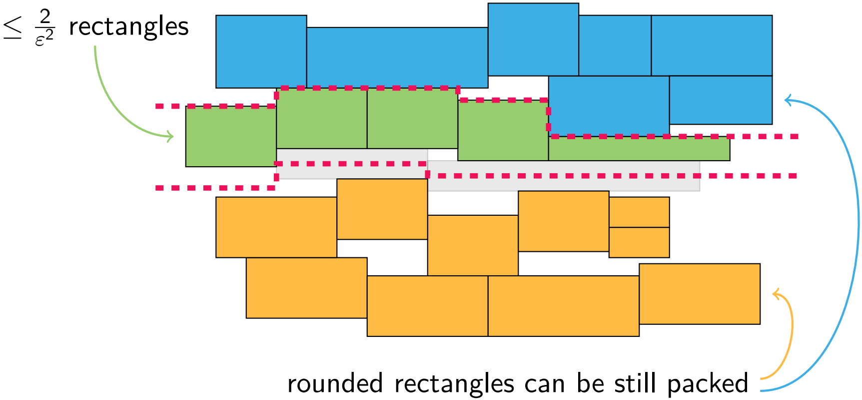
The structural lemma

If all rectangles have substantial width, by deleting εk rectangles:



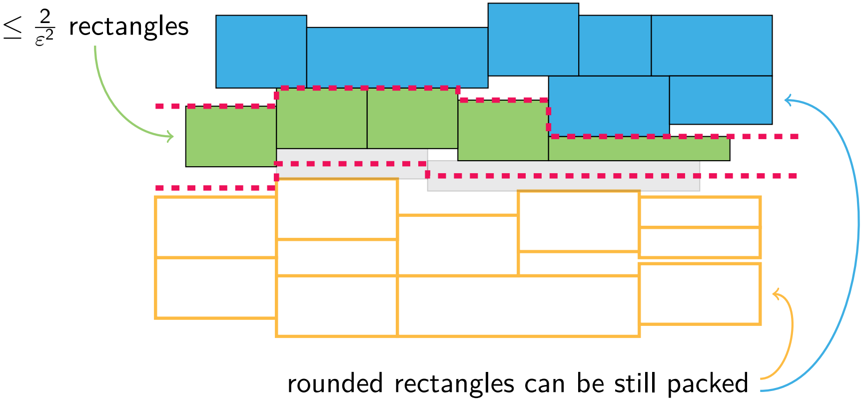
The structural lemma

If all rectangles have substantial width, by deleting εk rectangles:



The structural lemma

If all rectangles have substantial width, by deleting εk rectangles:



Brief aside: rounded rectangles

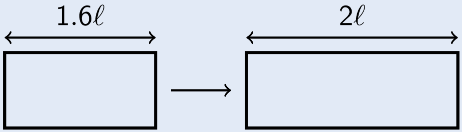
Definition

Rounding a rectangle to a multiple of $\ell = N_1/f(k,\delta)$.

Brief aside: rounded rectangles

Definition

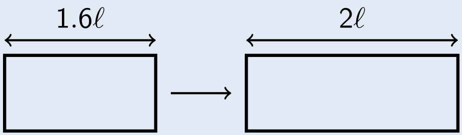
Rounding a rectangle to a multiple of $\ell = N_1/f(k,\delta)$.



Brief aside: rounded rectangles

Definition

Rounding a rectangle to a multiple of $\ell = N_1/f(k,\delta)$.



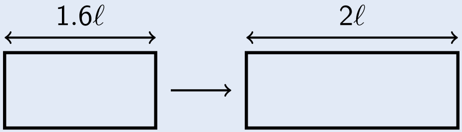
Properties

1. Interesting rectangles: have one of the k smallest height for their own width.

Brief aside: rounded rectangles

Definition

Rounding a rectangle to a multiple of $\ell = N_1/f(k,\delta)$.



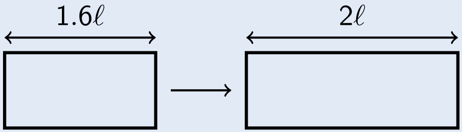
Properties

1. Interesting rectangles: have one of the k smallest height for their own width.
2. Bounded number of possible widths: $\leq N_1/\ell = f(k,\delta)$.

Brief aside: rounded rectangles

Definition

Rounding a rectangle to a multiple of $\ell = N_1/f(k,\delta)$.



Properties

1. Interesting rectangles: have one of the k smallest height for their own width.
2. Bounded number of possible widths: $\leq N_1/\ell = f(k,\delta)$.
3. Bounded amount of interesting rounded rectangles $\Longrightarrow$ brute force.

Putting it all together

1. Remove ≤ 1 rectangle: reduction to rectangles of substantial width.

Putting it all together

1. Remove ≤ 1 rectangle: reduction to rectangles of substantial width.
2. DP on possible regions: find a structured solution of size $(1-\varepsilon)k$.

Putting it all together

1. Remove ≤ 1 rectangle: reduction to rectangles of substantial width.
2. DP on possible regions: find a structured solution of size $(1-\varepsilon)k$.
3. Inside each region: solve independently with color-coding, to avoid repetition between regions.

Putting it all together

1. Remove ≤ 1 rectangle: reduction to rectangles of substantial width.
2. DP on possible regions: find a structured solution of size $(1-\varepsilon)k$.
3. Inside each region: solve independently with color-coding, to avoid repetition between regions.
4. You've found a good solution ☺

Putting it all together

1. Remove ≤ 1 rectangle: reduction to rectangles of substantial width.
2. DP on possible regions: find a structured solution of size $(1-\varepsilon)k$.
3. Inside each region: solve independently with color-coding, to avoid repetition between regions.
4. You've found a good solution ☺

Theorem

2D-KNAPSACK *admits a PAS under the following assuptions:*

- ≤ *Unary setting*
- ↔ *Every rectangle is wide*
- 📐 *Box aspect ratio is bounded*

Conclusion

- PAS under assumptions (unary setting, wide, bounded aspect ratio)

Conclusion

- PAS under assumptions (unary setting, wide, bounded aspect ratio)
- Open question:
  ❓ Find a PAS without the wide assumption

Conclusion

- PAS under assumptions (unary setting, wide, bounded aspect ratio)
- Open question:
  ? Find a PAS without the wide assumption
- Wide assumption necessary!
  ⚠ Guarantees reduction to rectangles of substantial width.

Thank You

Thanks for listening!