# Towards the 5/6-Density Conjecture of Pinwheel Scheduling

Ben Smith
Leszek Gąsieniec and Sebastian Wild

University of Liverpool

January 2022

# Outline

Introduction

Single instances

Pareto surfaces

Returning to 5/6

# Outline

Introduction

Single instances

Pareto surfaces

Returning to 5/6

# Introduction – the problem

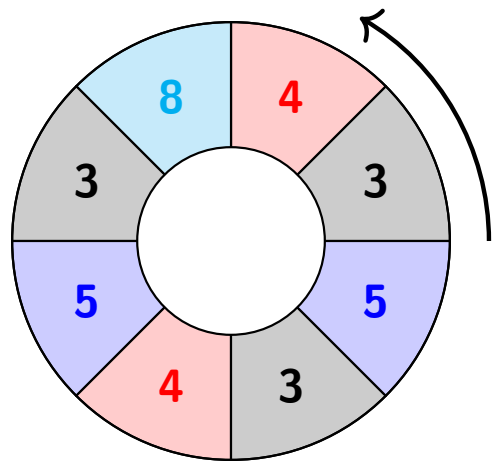
- The Pinwheel Scheduling Problem[1]:
  - Given $k$ tasks (each with *frequency* $a_i \in \mathbb{N}$)
    Is there a perpetual schedule for a (single) agent who can perform one task per day such that every section of length $a_i$ contains at least one instance of task $i$?

[1] Holte et.al., 1989
[2] Jacobs and Longo, 2014(arxiv)

# Introduction – the problem

- The Pinwheel Scheduling Problem[1]:
  - Given $k$ tasks (each with *frequency* $a_i \in \mathbb{N}$)
    Is there a perpetual schedule for a (single) agent who can perform one task per day such that every section of length $a_i$ contains at least one instance of task $i$?
- Example:
  - $A = \underbrace{[3, 4, 5, 8]}_{k=4}$
  - $S = 1, 2, 4, 1, 3, 2, 1, 3$

| Task | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Frequency | 3 | 4 | 5 | 8 |

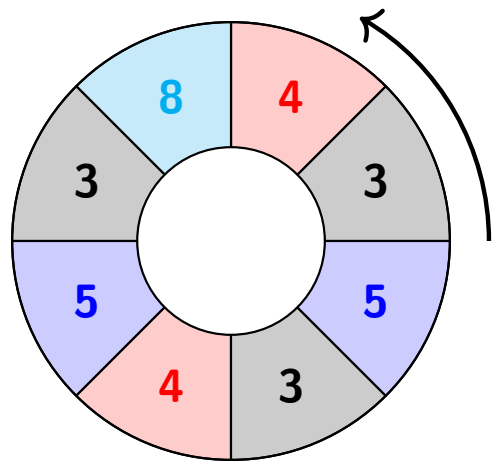



[1] Holte et.al., 1989
[2] Jacobs and Longo, 2014(arxiv)

# Introduction – the problem

- The Pinwheel Scheduling Problem[1]:
  - Given $k$ tasks (each with *frequency* $a_i \in \mathbb{N}$)
    Is there a perpetual schedule for a (single) agent who can perform one task per day such that every section of length $a_i$ contains at least one instance of task $i$?
- Example:
  - $A = \underbrace{[3, 4, 5, 8]}_{k=4}$
  - $S = 1, 2, 4, 1, 3, 2, 1, 3$
  - NP Hard[2]

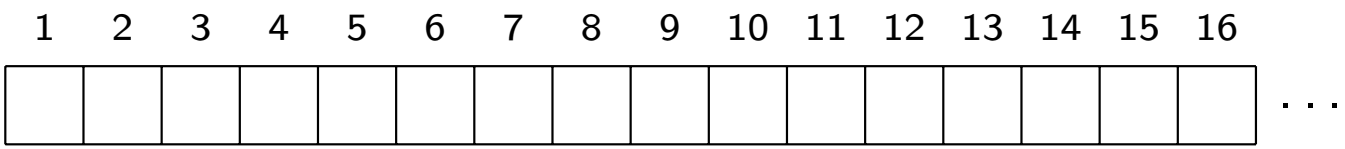
| Task | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Frequency | 3 | 4 | 5 | 8 |




[1] Holte et.al., 1989
[2] Jacobs and Longo, 2014(arxiv)

# Introduction – the 5/6 Density Conjecture

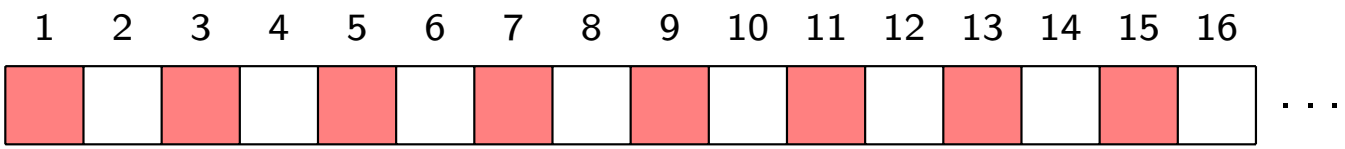
- Density $d(A) = \sum_{i=1}^{k} 1/a_i$
- $[2, 3, M]$




[3] Chan and Chin, 1993
[4] Ding, 2020

# Introduction – the 5/6 Density Conjecture

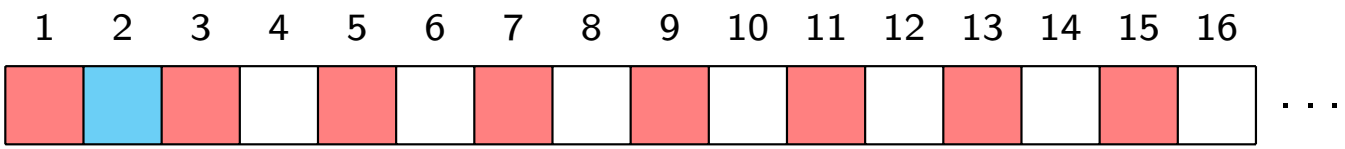
- Density $d(A) = \sum_{i=1}^{k} 1/a_i$
- $[2, 3, M]$




[3] Chan and Chin, 1993
[4] Ding, 2020

# Introduction – the 5/6 Density Conjecture

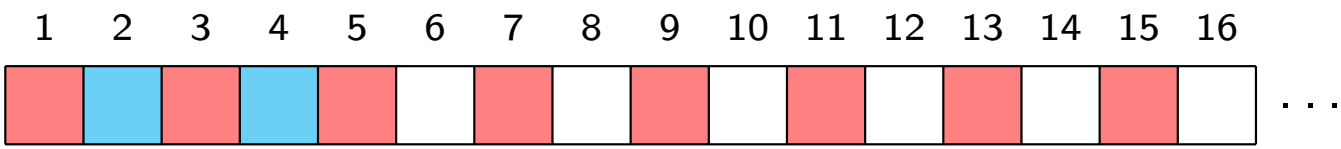
- Density $d(A) = \sum_{i=1}^{k} 1/a_i$
- $[2, 3, M]$




[3] Chan and Chin, 1993
[4] Ding, 2020

# Introduction – the 5/6 Density Conjecture

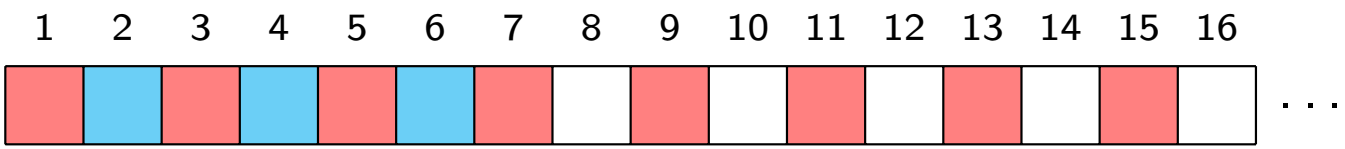
- Density $d(A) = \sum_{i=1}^{k} 1/a_i$
- $[2, 3, M]$




[3] Chan and Chin, 1993
[4] Ding, 2020

# Introduction – the 5/6 Density Conjecture

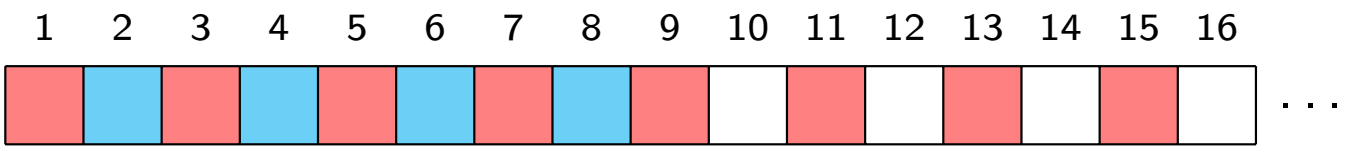
- Density $d(A) = \sum_{i=1}^{k} 1/a_i$
- $[2, 3, M]$




[3] Chan and Chin, 1993
[4] Ding, 2020

# Introduction – the 5/6 Density Conjecture

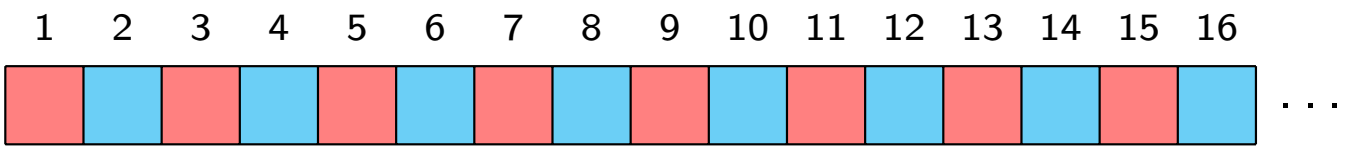
- Density $d(A) = \sum_{i=1}^{k} 1/a_i$
- $[2, 3, M]$




[3] Chan and Chin, 1993
[4] Ding, 2020

# Introduction – the 5/6 Density Conjecture

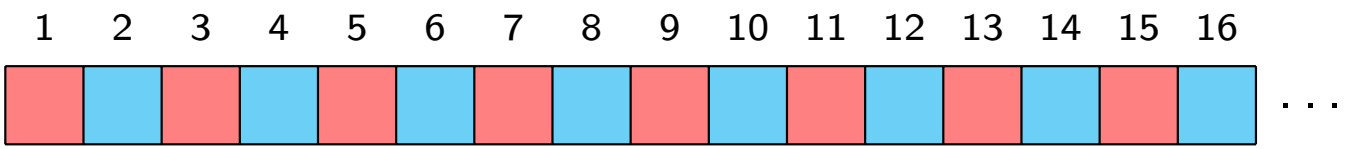
- Density $d(A) = \sum_{i=1}^{k} 1/a_i$
- $[2, 3, M]$




[3] Chan and Chin, 1993
[4] Ding, 2020

# Introduction – the 5/6 Density Conjecture

- Density $d(A) = \sum_{i=1}^{k} 1/a_i$
- $[2, 3, M]$

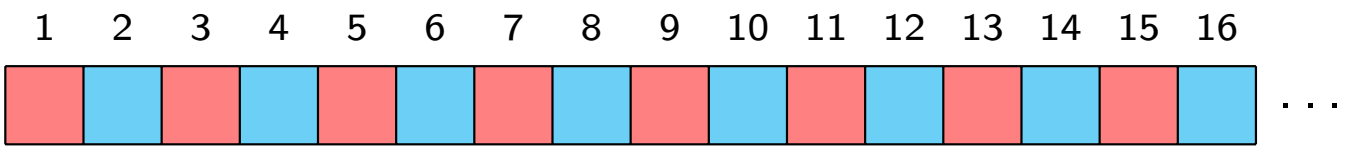



- **5/6 Density Conjecture:**
  All Pinwheel scheduling instances where $d \leq 5/6$ are schedulable.[3]

[3] Chan and Chin, 1993
[4] Ding, 2020

# Introduction – the 5/6 Density Conjecture

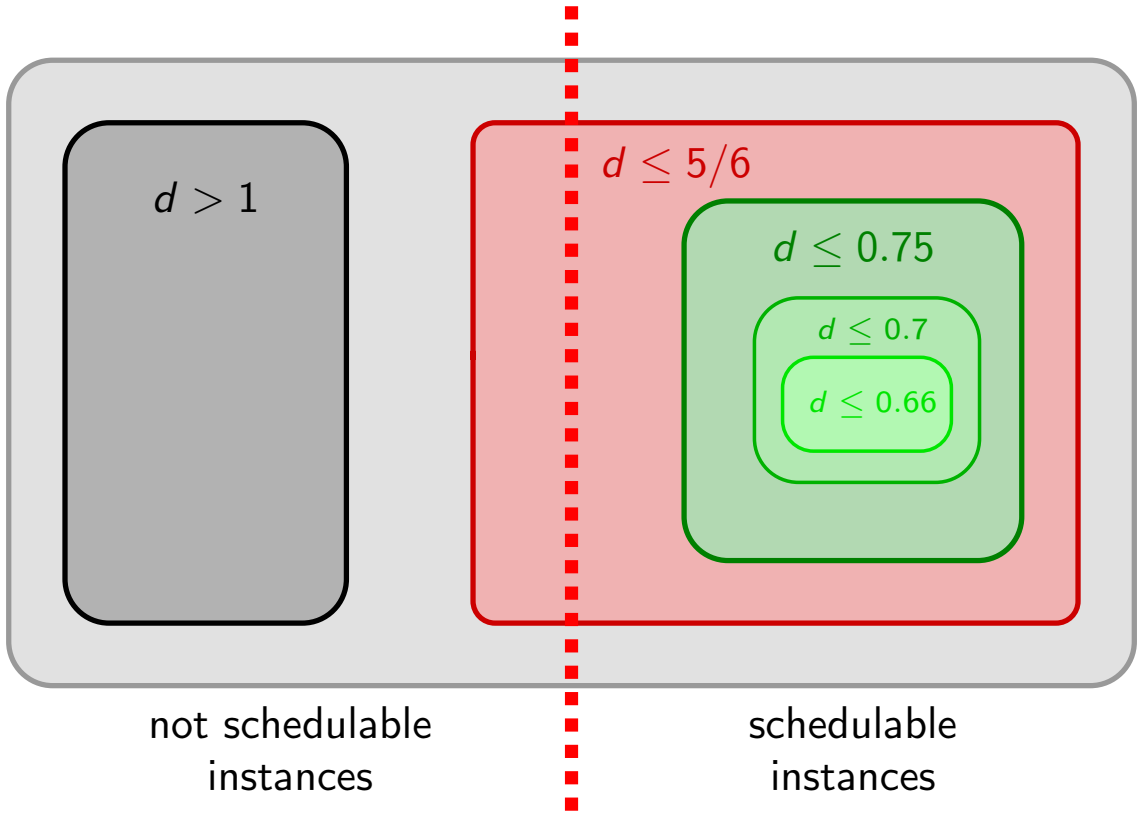
- Density $d(A) = \sum_{i=1}^{k} 1/a_i$
- $[2, 3, M]$



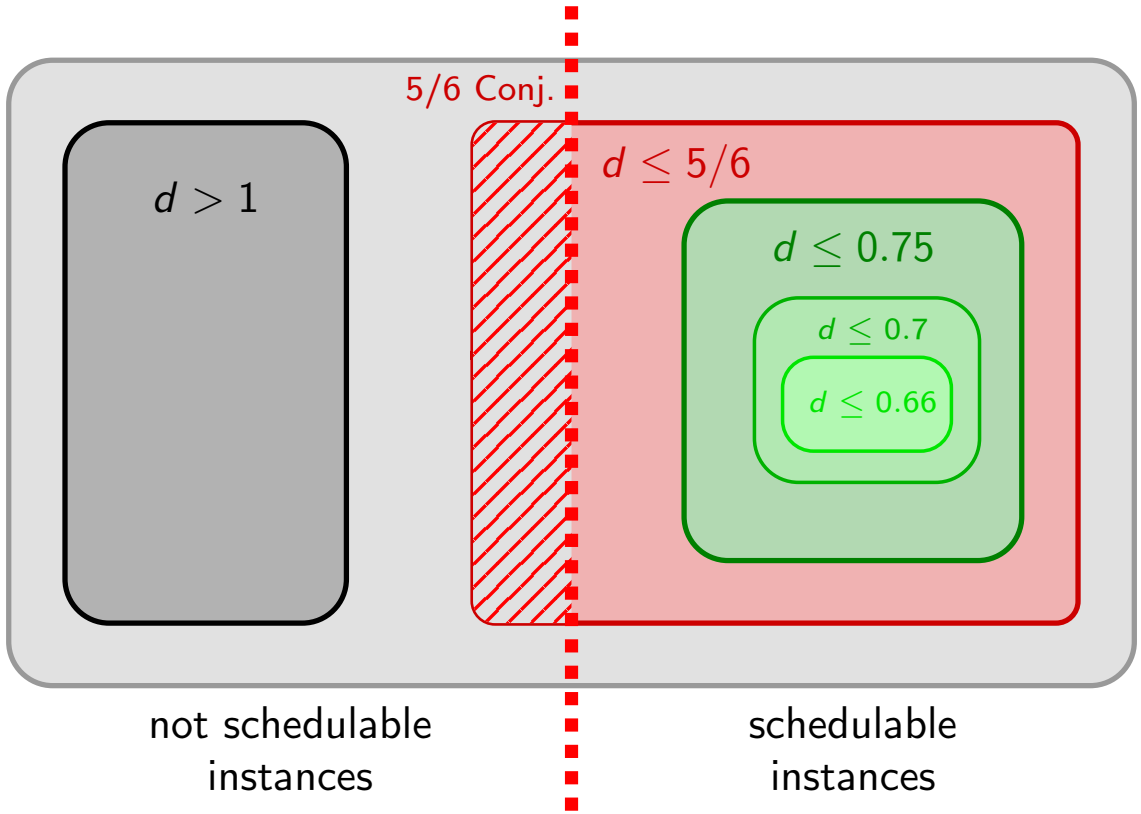

- **5/6 Density Conjecture:**
  All Pinwheel scheduling instances where $d \leq 5/6$ are schedulable.[3]
- Proven for $k \leq 5$ tasks.[4]
- **Contribution**: We prove this for $k \leq 12$

[3] Chan and Chin, 1993
[4] Ding, 2020

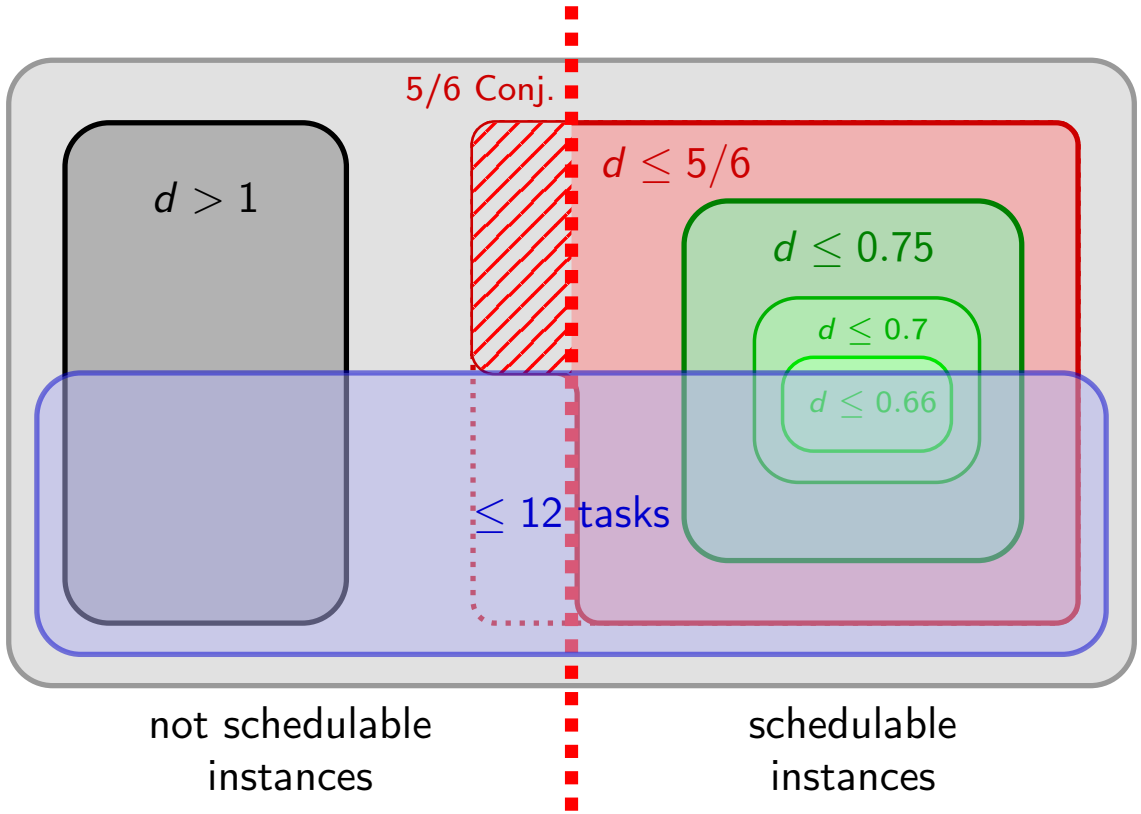
# Introduction – the literature



# Introduction – the literature



# Introduction – the literature



# Outline

Introduction

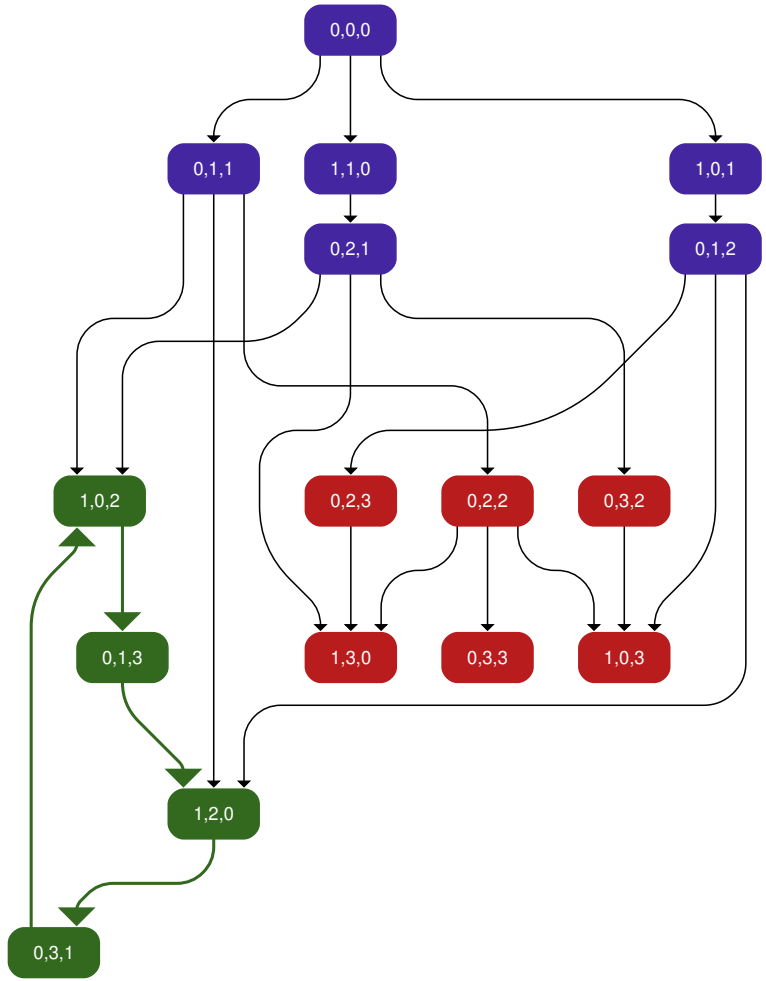
Single instances

Pareto surfaces

Returning to 5/6

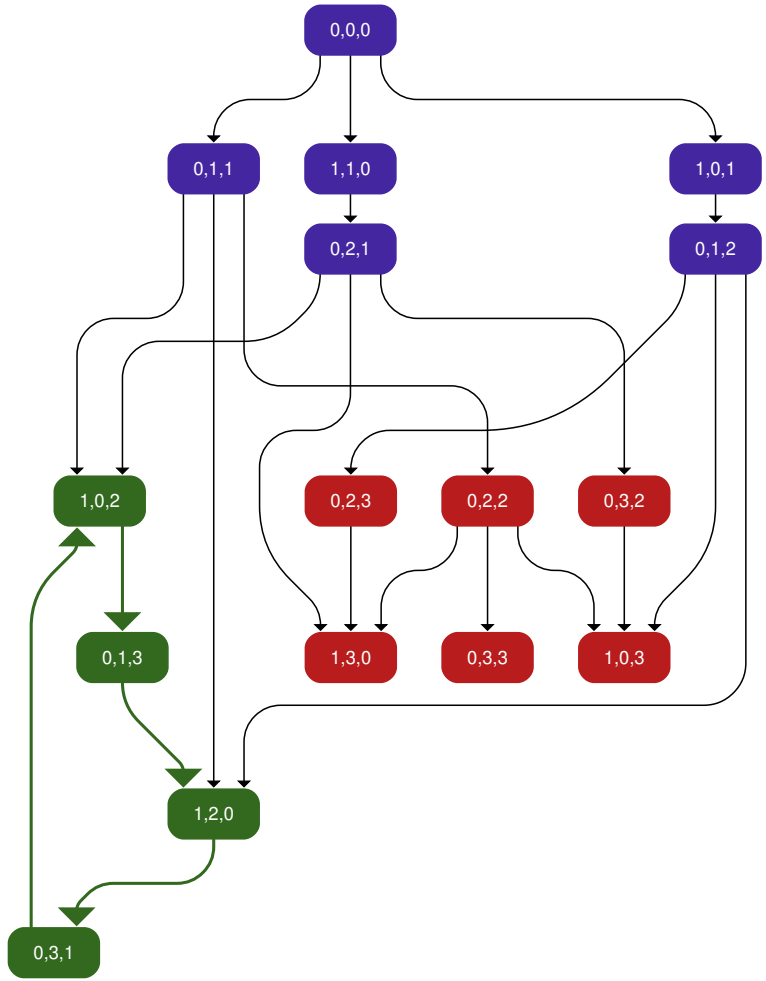
# Single instances

- Pinwheel as a directed graph
  - Example: $[2, 4, 4]$
- Schedulable iff the graph contains a directed cycle
- Slow, memory intensive – so optimize the approach



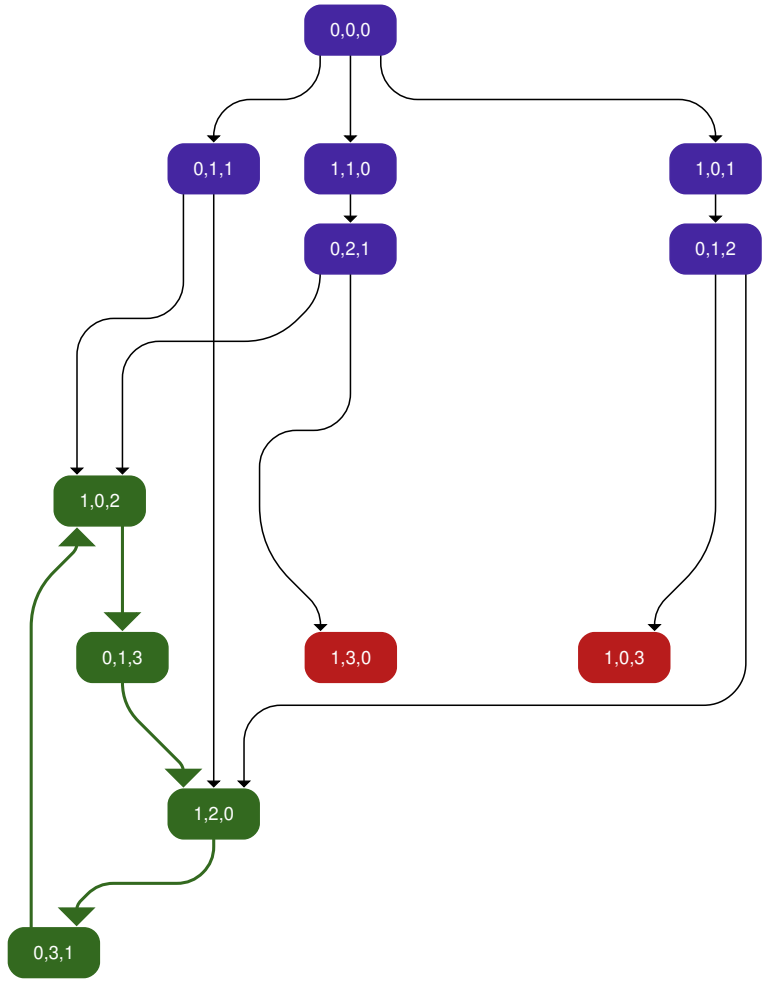
# Single instances – Optimisations

1. Naïve
   - Generate the graph on demand as a DFS tree



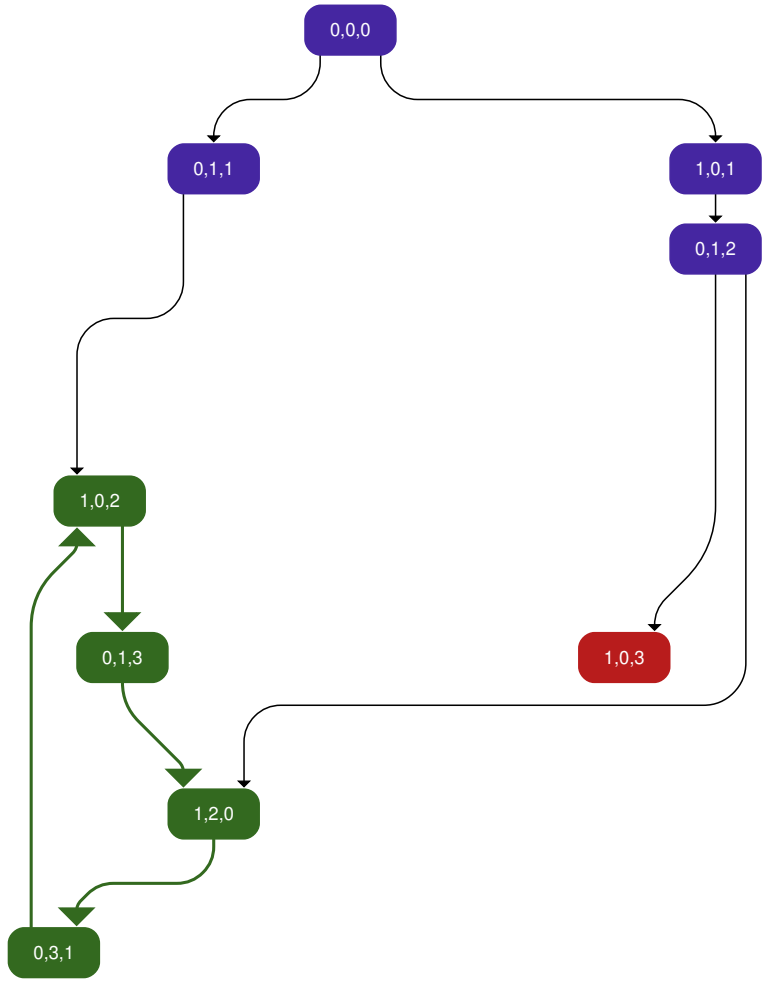
# Single instances – Optimisations

1. Naïve
   - Generate the graph on demand as a DFS tree
2. Optimised
   - Forbid immediate repetition



# Single instances – Optimisations

1. Naïve
   - Generate the graph on demand as a DFS tree
2. Optimised
   - Forbid immediate repetition
   - Exploit symmetry between duplicated frequencies
   - Limit testing using minimum solution length



# Single instances – Optimisations

1. Naïve
   - Generate the graph on demand as a DFS tree
2. Optimised
   - Forbid immediate repetition
   - Exploit symmetry between duplicated frequencies
   - Limit testing using minimum solution length
3. Foresight
   - *Urgency*: time until the next constraint violation
   - *Forcing*: finding the composition of the next few moves
   - *Dooming*: finding that a node must lead to constraint violations

- Consider a sample state from $[3, 4, 5, 8]$

1,2,4,0

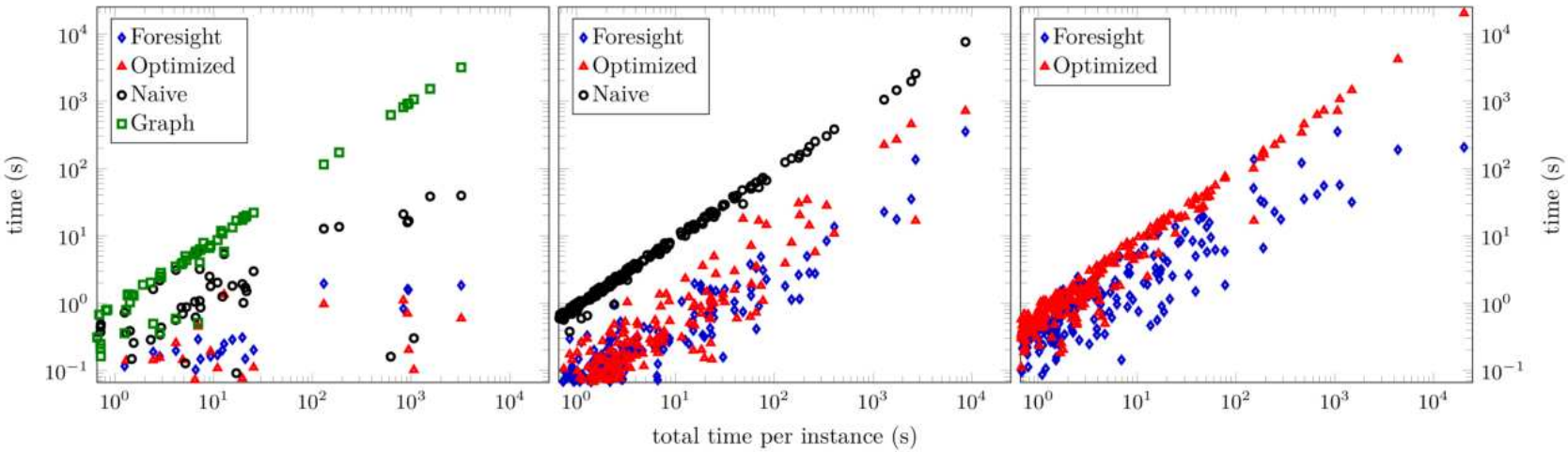
- Switch to Urgency

1,1,0,7

- $u_2 = 0 \rightarrow u_2$ is forced
- 3 elements with $u_i \leq 1 \rightarrow$ this state is doomed

1,1,0,7

# Single instances – Tournament of Random Instances




- "hardness" of input is challenging to predict
- $x$-axis is measured time to solve, not input size

# Outline

Introduction

Single instances

Pareto surfaces

Returning to 5/6

# Pareto surfaces

**Idea:** Find set $\mathcal{C}$ of schedules that solves entire family of instances.

i.e. an instance is schedulable $\iff$ a schedule from $\mathcal{C}$ solves it

**Theorem:** Any set of schedulable PWS instances with $k$ tasks can be solved by a finite set of schedules $\mathcal{C}_k$.

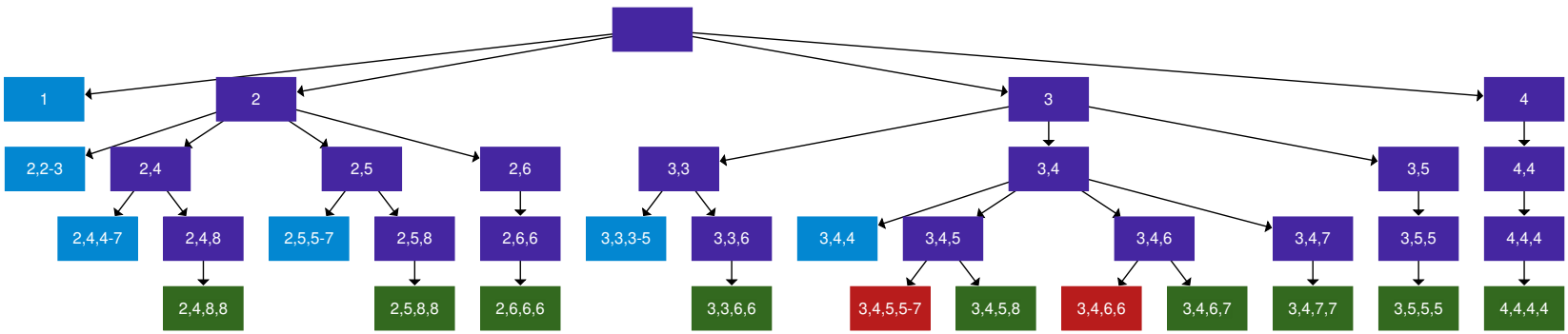
# Pareto surfaces

**Idea:** Find set $\mathcal{C}$ of schedules that solves entire family of instances.

i.e. an instance is schedulable $\iff$ a schedule from $\mathcal{C}$ solves it

**Theorem:** Any set of schedulable PWS instances with $k$ tasks can be solved by a finite set of schedules $\mathcal{C}_k$.

Proof yields an algorithm to find $\mathcal{C}_k$ by computing the *Pareto Trie*:




$\rightsquigarrow$ every instance with $k = 4$ is either:

- written in the trie
- an unschedulable child of a blue node
- shares a schedule with a green node

# Pareto surfaces up to $k = 5$

| Instance | Schedule |
|---|---|
| $(1)$ | $(1)$ |
| $(2, 2)$ | $(1, 2)$ |
| $(2, 4, 4)$ | $(1, 2, 1, 3)$ |
| $(3, 3, 3)$ | $(1, 2, 3)$ |
| $(2, 4, 8, 8)$ | $(1, 2, 1, 3, 1, 2, 1, 4)$ |
| $(2, 6, 6, 6)$ | $(1, 2, 1, 3, 1, 4)$ |
| $(3, 3, 6, 6)$ | $(1, 2, 3, 1, 2, 4)$ |
| $(3, 4, 5, 8)$ | $(1, 2, 4, 1, 3, 2, 1, 3)$ |
| $(3, 5, 5, 5)$ | $(1, 2, 3, 1, 4, 2, 1, 3, 4)$ |
| $(4, 4, 4, 4)$ | $(1, 2, 3, 4)$ |

| Instance | Schedule |
|---|---|
| $(2, 4, 8, 16, 16)$ | $(1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5)$ |
| $(2, 4, 12, 12, 12)$ | $(1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 5)$ |
| $(2, 6, 6, 12, 12)$ | $(1, 2, 1, 3, 1, 4, 1, 2, 1, 3, 1, 5)$ |
| $(2, 6, 8, 10, 16)$ | $(1, 2, 1, 3, 1, 5, 1, 2, 1, 4, 1, 3, 1, 2, 1, 4)$ |
| $(2, 6, 10, 10, 10)$ | $(1, 2, 1, 3, 1, 4, 1, 2, 1, 5, 1, 3, 1, 2, 1, 4, 1, 5)$ |
| $(2, 8, 8, 8, 8)$ | $(1, 2, 1, 3, 1, 4, 1, 5)$ |
| $(3, 3, 6, 12, 12)$ | $(1, 2, 3, 1, 2, 4, 1, 2, 3, 1, 2, 5)$ |
| $(3, 3, 9, 9, 9)$ | $(1, 2, 3, 1, 2, 4, 1, 2, 5)$ |
| $(3, 4, 5, 14, 14)$ | $(1, 2, 3, 1, 4, 2, 1, 3, 1, 2, 5, 1, 3, 2)$ |
| $(3, 4, 6, 10, 16)$ | $(1, 2, 3, 1, 4, 2, 1, 3, 1, 2, 4, 1, 3, 2, 1, 5)$ |
| $(3, 4, 6, 11, 11)$ | $(1, 2, 3, 1, 5, 2, 1, 3, 2, 1, 4)$ |
| $(3, 4, 8, 8, 8)$ | $(1, 2, 4, 1, 5, 2, 1, 3)$ |
| $(3, 5, 5, 9, 9)$ | $(1, 2, 5, 1, 3, 2, 1, 4, 3)$ |
| $(3, 5, 6, 7, 12)$ | $(1, 2, 4, 1, 3, 2, 1, 4, 2, 1, 3, 5)$ |
| $(3, 5, 7, 7, 9)$ | $(1, 2, 3, 1, 4, 2, 1, 5, 3, 1, 2, 4, 1, 3, 2, 1, 5, 4)$ |
| $(3, 5, 7, 8, 8)$ | $(1, 2, 3, 1, 4, 2, 1, 5, 1, 3, 2, 1, 4, 5)$ |
| $(3, 6, 6, 6, 6)$ | $(1, 2, 3, 1, 4, 5)$ |
| $(4, 4, 4, 8, 8)$ | $(1, 2, 3, 4, 1, 2, 3, 5)$ |
| $(4, 4, 5, 7, 12)$ | $(1, 2, 3, 4, 1, 2, 5, 3, 1, 2, 4, 3)$ |
| $(4, 4, 6, 6, 6)$ | $(1, 3, 2, 4, 1, 5, 2, 3, 1, 4, 2, 5)$ |
| $(4, 5, 5, 6, 10)$ | $(1, 2, 3, 5, 1, 4, 2, 3, 1, 4)$ |
| $(4, 5, 5, 7, 7)$ | $(1, 2, 5, 3, 1, 4, 2, 1, 3, 5, 2, 1, 4, 3)$ |
| $(5, 5, 5, 5, 5)$ | $(1, 2, 3, 4, 5)$ |

# Pareto surfaces up to $k = 5$

| Instance | Schedule |
|---|---|
| $(1)$ | $(1)$ |
| $(2, 2)$ | $(1, 2)$ |
| $(2, 4, 4)$ | $(1, 2, 1, 3)$ |
| $(3, 3, 3)$ | $(1, 2, 3)$ |
| $(2, 4, 8, 8)$ | $(1, 2, 1, 3, 1, 2, 1, 4)$ |
| $(2, 6, 6, 6)$ | $(1, 2, 1, 3, 1, 4)$ |
| $(3, 3, 6, 6)$ | $(1, 2, 3, 1, 2, 4)$ |
| $(3, 4, 5, 8)$ | $(1, 2, 4, 1, 3, 2, 1, 3)$ |
| $(3, 5, 5, 5)$ | $(1, 2, 3, 1, 4, 2, 1, 3, 4)$ |
| $(4, 4, 4, 4)$ | $(1, 2, 3, 4)$ |

| Instance | Schedule |
|---|---|
| $(2, 4, 8, 16, 16)$ | $(1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5)$ |
| $(2, 4, 12, 12, 12)$ | $(1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 5)$ |
| $(2, 6, 6, 12, 12)$ | $(1, 2, 1, 3, 1, 4, 1, 2, 1, 3, 1, 5)$ |
| $(2, 6, 8, 10, 16)$ | $(1, 2, 1, 3, 1, 5, 1, 2, 1, 4, 1, 3, 1, 2, 1, 4)$ |
| $(2, 6, 10, 10, 10)$ | $(1, 2, 1, 3, 1, 4, 1, 2, 1, 5, 1, 3, 1, 2, 1, 4, 1, 5)$ |
| $(2, 8, 8, 8, 8)$ | $(1, 2, 1, 3, 1, 4, 1, 5)$ |
| $(3, 3, 6, 12, 12)$ | $(1, 2, 3, 1, 2, 4, 1, 2, 3, 1, 2, 5)$ |
| $(3, 3, 9, 9, 9)$ | $(1, 2, 3, 1, 2, 4, 1, 2, 5)$ |
| $(3, 4, 5, 14, 14)$ | $(1, 2, 3, 1, 4, 2, 1, 3, 1, 2, 5, 1, 3, 2)$ |
| $(3, 4, 6, 10, 16)$ | $(1, 2, 3, 1, 4, 2, 1, 3, 1, 2, 4, 1, 3, 2, 1, 5)$ |
| $(3, 4, 6, 11, 11)$ | $(1, 2, 3, 1, 5, 2, 1, 3, 2, 1, 4)$ |
| $(3, 4, 8, 8, 8)$ | $(1, 2, 4, 1, 5, 2, 1, 3)$ |
| $(3, 5, 5, 9, 9)$ | $(1, 2, 5, 1, 3, 2, 1, 4, 3)$ |
| $(3, 5, 6, 7, 12)$ | $(1, 2, 4, 1, 3, 2, 1, 4, 2, 1, 3, 5)$ |
| $(3, 5, 7, 7, 9)$ | $(1, 2, 3, 1, 4, 2, 1, 5, 3, 1, 2, 4, 1, 3, 2, 1, 5, 4)$ |
| $(3, 5, 7, 8, 8)$ | $(1, 2, 3, 1, 4, 2, 1, 5, 1, 3, 2, 1, 4, 5)$ |
| $(3, 6, 6, 6, 6)$ | $(1, 2, 3, 1, 4, 5)$ |
| $(4, 4, 4, 8, 8)$ | $(1, 2, 3, 4, 1, 2, 3, 5)$ |
| $(4, 4, 5, 7, 12)$ | $(1, 2, 3, 4, 1, 2, 5, 3, 1, 2, 4, 3)$ |
| $(4, 4, 6, 6, 6)$ | $(1, 3, 2, 4, 1, 5, 2, 3, 1, 4, 2, 5)$ |
| $(4, 5, 5, 6, 10)$ | $(1, 2, 3, 5, 1, 4, 2, 3, 1, 4)$ |
| $(4, 5, 5, 7, 7)$ | $(1, 2, 5, 3, 1, 4, 2, 1, 3, 5, 2, 1, 4, 3)$ |
| $(5, 5, 5, 5, 5)$ | $(1, 2, 3, 4, 5)$ |

**Observation:** All Pareto Surface members frequencies obey $a_i \leq 2^{k-1}$.

# Outline

Introduction

Single instances

Pareto surfaces

Returning to 5/6

# 5/6 Surface

- Find $\mathcal{C}_{k,d \leq 5/6}$

  i.e. Surface of instance with $k$ tasks and density $\leq 5/6$
- Optimize using the density gap & the belief that all instances are schedulable
  - "Unfolding"
  - "Folding"
  - Kernelization

    (also some data structures)

# 5/6 Surface – Unfolding

- Recall $A = [3, 4, 5, 8]$, and its schedule $S = 1, 2, 4, 1, 3, 2, 1, 3$
- Use this to solve $A' = [3, 4, 5, 16, 16]$ by *unfolding* $S$
- $S' = 1, 2, 4, 1, 3, 2, 1, 3$
  $1, 2, 5, 1, 3, 2, 1, 3$
- **Application**: Initializing the Pareto Surface
  - $153 \pm 2$ times less oracle consultations at $k = 8$

# 5/6 Surface – Folding

- Consider a challenging looking input:
  - $F_1 = [5, 6, 7, 9, 18, 19, 21]$, with $k = 7$ and $d = 0.776$

# 5/6 Surface – Folding

- Consider a challenging looking input:
  - $F_1 = [5, 6, 7, 9, 18, 19, 21]$, with $k = 7$ and $d = 0.776$
- Combine the last several elements to make problems whose unfolded schedules would solve $F_1$:

| | $k$ | $d$ |
|---|---|---|
| $F_2 = [5, 6, 7, 9, 9, 18]$ | 6 | 0.787 |
| $F_3 = [5, 6, 6, 7, 9]$ | 5 | 0.787 |
| $F_4 = [2, 5, 6, 7]$ | 4 | 1.001 |

# 5/6 Surface – Folding

- Consider a challenging looking input:
  - $F_1 = [5, 6, 7, 9, 18, \mathbf{19, 21}]$, with $k = 7$ and $d = 0.776$
- Combine the last several elements to make problems whose unfolded schedules would solve $F_1$:

| | $k$ | $d$ |
|---|---|---|
| $F_2 = [5, 6, 7, 9, \mathbf{9}, 18]$ | 6 | 0.787 |
| $F_3 = [5, 6, 6, 7, 9]$ | 5 | 0.787 |
| $F_4 = [2, 5, 6, 7]$ | 4 | 1.001 |

# 5/6 Surface – Folding

- Consider a challenging looking input:
  - $F_1 = [5, 6, 7, 9, \mathbf{18}, \mathbf{19}, \mathbf{21}]$, with $k = 7$ and $d = 0.776$
- Combine the last several elements to make problems whose unfolded schedules would solve $F_1$:

| | $k$ | $d$ |
|---|---|---|
| $F_2 = [5, 6, 7, 9, 9, 18]$ | 6 | 0.787 |
| $F_3 = [5, 6, \mathbf{6}, 7, 9]$ | 5 | 0.787 |
| $F_4 = [2, 5, 6, 7]$ | 4 | 1.001 |

# 5/6 Surface – Folding

- Consider a challenging looking input:
  - $F_1 = [5, 6, 7, \mathbf{9, 18, 19, 21}]$, with $k = 7$ and $d = 0.776$
- Combine the last several elements to make problems whose unfolded schedules would solve $F_1$:

| | $k$ | $d$ |
|---|---|---|
| $F_2 = [5, 6, 7, 9, 9, 18]$ | 6 | 0.787 |
| $F_3 = [5, 6, 6, 7, 9]$ | 5 | 0.787 |
| $F_4 = [\mathbf{2}, 5, 6, 7]$ | 4 | 1.001 |

# 5/6 Surface – Folding

- Consider a challenging looking input:
  - $F_1 = [5, 6, 7, 9, 18, 19, 21]$, with $k = 7$ and $d = 0.776$
- Combine the last several elements to make problems whose unfolded schedules would solve $F_1$:

| | $k$ | $d$ |
|---|---|---|
| $F_2 = [5, 6, 7, 9, 9, 18]$ | 6 | 0.787 |
| $F_3 = [5, 6, 6, 7, 9]$ | 5 | 0.787 |
| $F_4 = [2, 5, 6, 7]$ | 4 | 1.001 |

- **Observation 1**: $F_3$ is trivially schedulable by $S = 0, 1, 2, 3, 4$
- **Observation 2**: this can make unschedulable problems from schedulable inputs

# 5/6 Surface – Folding

- Consider a challenging looking input:
  - $F_1 = [5, 6, 7, 9, 18, 19, 21]$, with $k = 7$ and $d = 0.776$
- Combine the last several elements to make problems whose unfolded schedules would solve $F_1$:

| | $k$ | $d$ |
|---|---|---|
| $F_2 = [5, 6, 7, 9, 9, 18]$ | 6 | 0.787 |
| $F_3 = [5, 6, 6, 7, 9]$ | 5 | 0.787 |
| $F_4 = [2, 5, 6, 7]$ | 4 | 1.001 |

- **Observation 1**: $F_3$ is trivially schedulable by $S = 0, 1, 2, 3, 4$
- **Observation 2**: this can make unschedulable problems from schedulable inputs
- **Application**: Attempt multiple foldings in parallel

# 5/6 Surface – Kernelization

- **Kernalization Conjecture**: Any frequency $a_i \geq 2^{k-1}$ can be reduced to $a_i' = 2^{k-1}$ without affecting schedulability
- **Application**: Double down – test both conjectures simultaneously

# 5/6 Surface – Kernelization

- **Kernalization Conjecture**: Any frequency $a_i \geq 2^{k-1}$ can be reduced to $a_i' = 2^{k-1}$ without affecting schedulability
- **Application**: Double down – test both conjectures simultaneously
- **Achievement**: Kernalisation Conjecture true for $k \leq 12$ and $d \leq 5/6$
- **Achievement**: 5/6 Conjecture true for $k \leq 12$

# Conclusion

Results:

- The Pareto surface for $k = 5$.
- Proof of the 5/6 Conjecture for $k \leq 12$.
- The Kernalization Conjecture, which holds for $k \leq 5$ and $k \leq 12, d \leq 5/6$.
- A fast Pinwheel Scheduling oracle, available online.

# Conclusion

Results:

- The Pareto surface for $k = 5$.
- Proof of the 5/6 Conjecture for $k \leq 12$.
- The Kernalization Conjecture, which holds for $k \leq 5$ and $k \leq 12, d \leq 5/6$.
- A fast Pinwheel Scheduling oracle, available online.

Open problems:

- Implications for Bamboo Garden Trimming
- The Kernalization Conjecture
- The 5/6 Density Conjecture

Any questions?