Proof of the Density Threshold Conjecture for Pinwheel Scheduling

Akitoshi Kawamura (Kyoto)

June 28, 2024, Vancouver, BC, Canada

科研費
KAKENHI
JP20H00587
JP23K28036

henceforth $a_1 \le a_2 \le \cdots \le a_k$

Pinwheel scheduling [HMRTV89]

Each task $i = 1, \dots, k$ must be done **at least once in any a_i consecutive days**.
Can we achieve this by doing one task every day?

If we can, the k-tuple $A = (a_1, \dots, a_k)$ is said to be **schedulable**.

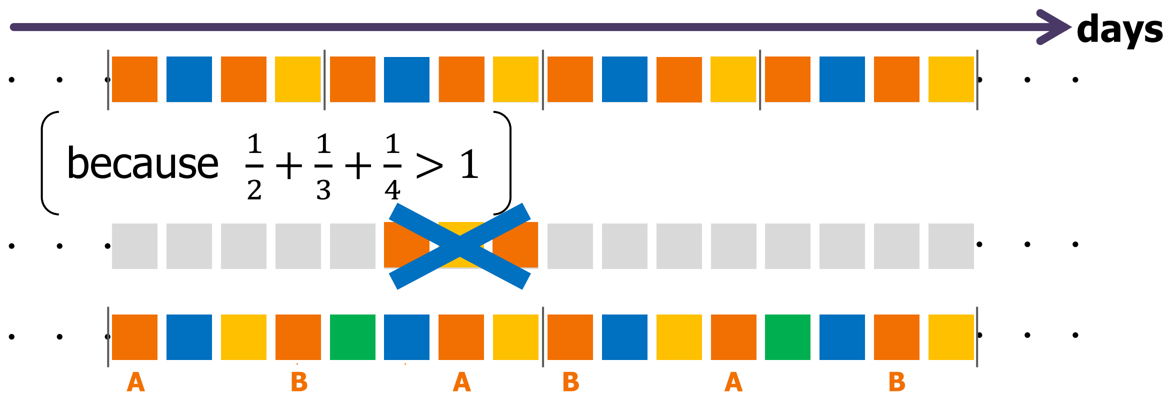
Examples

$(2, 4, 4)$ YES

$(2, 3, 4)$ NO because $\frac{1}{2} + \frac{1}{3} + \frac{1}{4} > 1$

$(2, 3, 100)$ NO

$(3, 4, 5, 8)$ YES



$(3, 4, 6, 8)$ YES

$(4, 5, 6, 6, 8)$ YES

A B

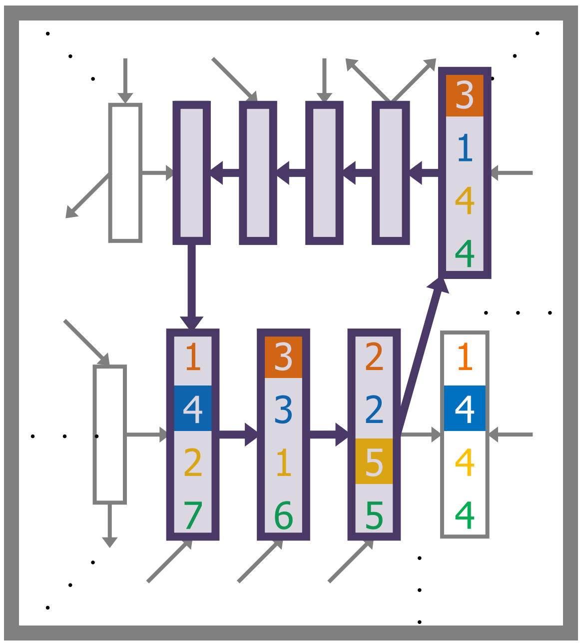
Schedulability is preserved by

RELAX replacing a number with a bigger one

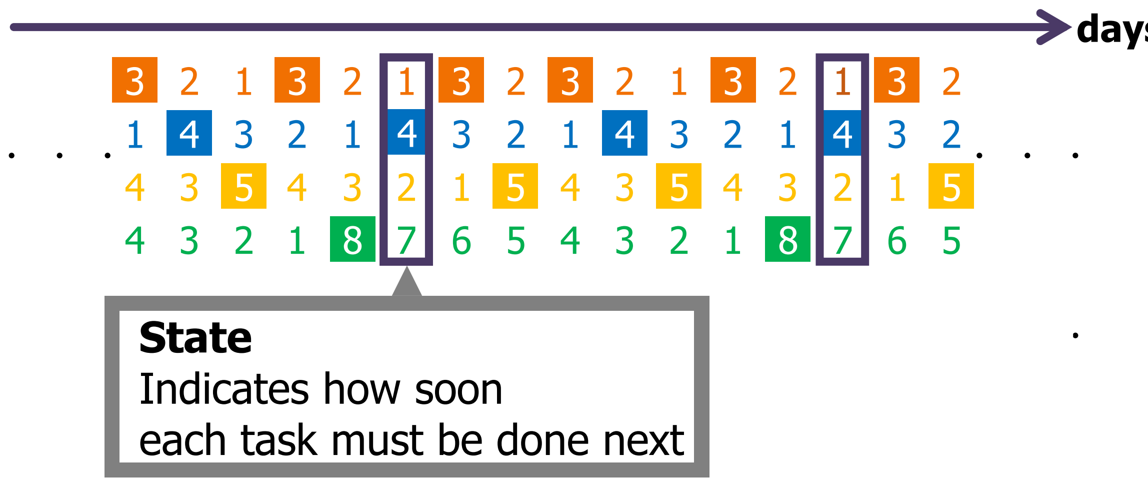
SPLIT replacing a number with two copies of its double (or N copies of its N-fold)

[HMRTV89] R. Holte, A. Mok, L. Rosier, I. Tulchinsky, D. Varvel. The pinwheel: a real-time scheduling problem. In Proc. 22nd Hawaii International Conference on System Science, pp. 693–702, 1989.

Schedulability can be tested in **PSPACE**:
Build the state transition graph and check for a cycle.



Graph for $(3, 4, 5, 8)$
(with $\leq 3 \cdot 4 \cdot 5 \cdot 8$ vertices)



Definition

The **density** of $A = (a_1, \dots, a_k)$ is $\mathrm{D}(A) = \frac{1}{a_1} + \cdots + \frac{1}{a_k}$.

For A to be schedulable, $\mathrm{D}(A) \le 1$ is necessary.
This is not always sufficient, but:

Can we make this number bigger?

Theorem [HMRTV89]

A is schedulable if $\mathrm{D}(A) \le 1$ and each number in A divides the next.

Proof: Reduce to a single kind of tasks.

Example $(4, 8, 8, 16, 16) \xleftarrow{\text{SPLIT}} (4, 8, 8, 8) \xleftarrow{\text{SPLIT}} (4, 4, 4)$

YES YES YES

Theorem [HRTV92]

A is schedulable if $\mathrm{D}(A) \le 1$ and A has only two distinct numbers.

Corollary [HMRTV89]

A is schedulable if $\mathrm{D}(A) \le \frac{1}{2}$.

Proof: Round down to powers of 2.

Example $(6, 12, 13, 24, 25) \xleftarrow{\text{RELAX}} (4, 8, 8, 16, 16)$

YES YES

Using these theorems more cleverly, we get "thriftier" rounding-down methods that improve the corollary (next page)

[HMRTV89] R. Holte, A. Mok, L. Rosier, I. Tulchinsky, D. Varvel. The pinwheel: a real-time scheduling problem. In Proc. 22nd Hawaii International Conference on System Science, pp. 693–702, 1989.
[HRTV92] R. Holte, L. Rosier, I. Tulchinsky, D. Varvel. Pinwheel scheduling with two distinct numbers. Theoretical Computer Science 100, 105–135, 1992.

Theorem (restated) [HMRTV89]
A is schedulable if $\mathrm{D}(A) \le \frac{1}{2}$.
$(= 0.5)$

Theorem [CC93]
A is schedulable if $\mathrm{D}(A) \le \frac{2}{3}$.
$(= 0.666\ldots)$

Theorem [CC92]
A is schedulable if $\mathrm{D}(A) \le \frac{7}{10}$.
$(= 0.7)$

Theorem [FL02]
A is schedulable if $\mathrm{D}(A) \le \frac{3}{4}$.
$(= 0.75)$

Theorem (this talk)
~~Conjecture~~ [CC93]
A is schedulable if $\mathrm{D}(A) \le \frac{5}{6}$.
$(= 0.833\ldots)$

(Best possible, because of $(2, 3, \bullet)$)

Partial progress:

Theorem [LL97]
Conjecture is true when A has only three distinct numbers.

Theorem [FL02]
Conjecture is true when $a_1 = 2$.

Theorem [GSW22]
Conjecture is true when $k \le 12$.

involve computers

[CC92] M.Y. Chan, F. Chin. General schedulers for the pinwheel problem based on double-integer reduction. IEEE Transactions on Computers 41, 755–768, 1992.
[CC93] M.Y. Chan, F. Chin. Schedulers for larger classes of pinwheel instances. Algorithmica 9, 425–462, 1993.
[FL02] P.C. Fishburn, J.C. Lagarias. Pinwheel scheduling: achievable densities. Algorithmica 34, 14–38, 2002.
[GSW22] L. Gąsieniec, B. Smith and S. Wild. Towards the 5/6-density conjecture of pinwheel scheduling. In Proc. SIAM Symposium on Algorithm Engineering and Experiments (ALENEX), pp. 91–103, 2022.
[HMRTV89] R. Holte, A. Mok, L. Rosier, I. Tulchinsky, D. Varvel. The pinwheel: a real-time scheduling problem. In Proc. 22nd Hawaii International Conference on System Science, pp. 693–702, 1989.
[LL97] S. Lin, K. Lin. A pinwheel scheduler for three distinct numbers with a tight schedulability bound. Algorithmica 19, 411–426, 1997.

Main Theorem (restated)

A is schedulable if $\mathrm{D}(A) \le \frac{5}{6}$.

Lemma

Verified by computer
(using transition graphs)

B is schedulable if $\mathrm{D}'(B) < \frac{5}{6} + \frac{1}{22}$ and B consists only of elements < 22.

Modified density D', defined by regarding $11, \ldots, 21$ as $12, \ldots, 22$

$$A = (2, 5, 15, 40, 51, 87)$$
$$\downarrow \text{①}$$
$$(2, 5, 15, 40, \underline{51, 51})$$
$$\downarrow \text{②}$$
$$(2, 5, 15, 25, 40)$$
$$\downarrow \text{①}$$
$$(2, 5, 15, \underline{25, 25})$$
$$\downarrow \text{②}$$
$$B = (2, 5, 12, 15)$$

$$\mathrm{D}'(B) = \frac{1}{2} + \frac{1}{5} + \frac{1}{13} + \frac{1}{16}$$

Proof of Theorem, assuming Lemma

Suppose that A was unschedulable.

Let B be the result of repeatedly applying ①② to A until all elements are ≤ 22.

① If we have a single biggest element, decrease it.

② If we have two copies of the biggest element, say a, a, replace them by one $\left\lfloor \frac{a}{2} \right\rfloor$.

Then B is unschedulable and $\mathrm{D}'(B) < \mathrm{D}(A) + \frac{1}{22} \le \frac{5}{6} + \frac{1}{22}$, contradicting the Lemma.

Because ①② are inverse operations of RELAX SPLIT

Main

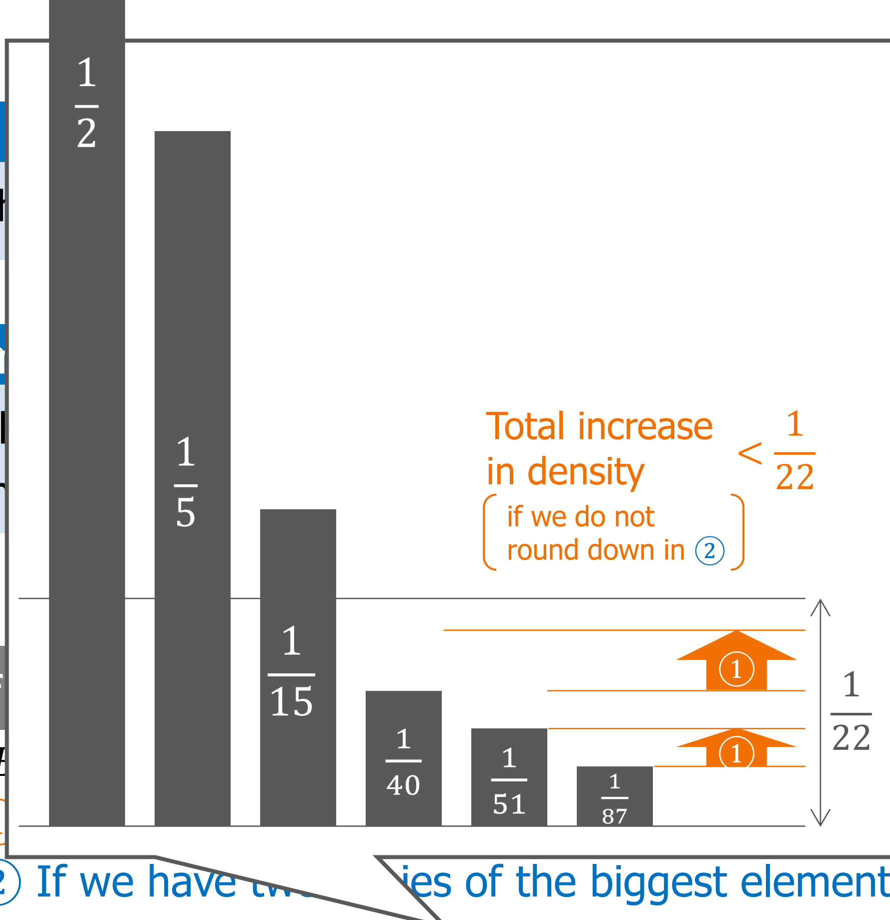
A is sch

Lemm

B is sch ... of

elemen ... by

, 22



$$A = (2, 5, 15, 40, 51, 87)$$

1

$$(2, 5, 15, 40, 51, 51)$$

2

$$(2, 5, 15, 25, 40)$$

1

$$(2, 5, 15, 25, 25)$$

2

$$B = (2, 5, 12, 15)$$

$$\mathrm{D}'(B) = \frac{1}{2} + \frac{1}{5} + \frac{1}{13} + \frac{1}{16}$$

Proof

... hat A was unschedulable.

Let B ... A until all elements are ≤ 22.

1 ... e it.

2 If we have tw... ies of the biggest element, say a, a, replace them by one $\left\lceil \frac{a}{2} \right\rceil$.

Then B is unschedulable and $\mathrm{D}'(B) < \mathrm{D}(A) + \frac{1}{22} \leq \frac{5}{6} + \frac{1}{22}$, contradicting the Lemma.

Because 1 2 are inverse operations of RELAX SPLIT

Future work

Proof of density bounds without brute-force search

Complexity of deciding schedulability

- Is it in $\mathbf{NP}$? (Note: some instances admit only super-polynomially long solutions)

  We know it is in $\mathbf{PSPACE}$, by the method of transition graphs

- Is it $\mathbf{NP}$-hard?

  We know it is $\mathbf{NP}$-hard if we can write "l copies of a", with l in binary, in the input

The covering version ("point patrolling" [KS20])

What if you can schedule task i **at most** once per a_i days (and must cover all days)?

- Lowest density guarantee lies somewhere between $1.264\ldots$ and 1.4125

Optimization and more applied versions of the problem

- Bamboo garden trimming [GJKLLMR24]:
  Minimize the maximum factor by which we violate the frequency requirements
- Implications for various applied settings

[GJKLLMR24] L. Gąsieniec, T. Jurdziński, R. Klasing, C. Levcopoulos, A. Lingas, J. Min, T. Radzik. Perpetual maintenance of machines with different urgency requirements. Journal of Computer and System Sciences 139, 103476, 2024.

[KS20] A. Kawamura, M. Soejima. Simple strategies versus optimal schedules in multi-agent patrolling. Theoretical Computer Science 839, 195–206, 2020.