An Optimal Density Bound for Discretized Point Patrolling

Ahan Mishra*

1 Context This document is a slice of a paper that I wrote with the title above. The full version is available on arXiv and the conference website. The purpose of the original paper was to communicate new results on problems studied in the literature along with the techniques needed for those results. This document has a more focused purpose and doesn't fully prove the titular result. The audience for the original paper was general theoretical computer scientists. This document should be more accessible and be fine for any technical audience with some math background. Note that because this is only part of the paper there may be references to content that is not there (i.e. cited papers and appendices), that content should not be strictly needed but it can be found in the full paper versions linked above.

Abstract. The pinwheel problem is a real-time scheduling problem that asks, given n tasks with periods $a_i \in \mathbb{N}$, whether it is possible to infinitely schedule the tasks, one per time unit, such that every task i is scheduled in every interval of a_i units. We study a corresponding version of this packing problem in the covering setting, stylized as the discretized point patrolling problem in the literature. Specifically, given n tasks with periods a_i, the problem asks whether it is possible to assign each day to a task such that every task i is scheduled at *most* once every a_i days. The density of an instance in either case is defined as the sum of the inverses of task periods. Recently, the long-standing 5/6 density bound conjecture in the packing setting was resolved affirmatively. The resolution means any instance with density at least 5/6 is schedulable. A corresponding conjecture was made in the covering setting and renewed multiple times in more recent work. We resolve this conjecture affirmatively by proving that every discretized point patrolling instance with density at least $\sum_{i=0}^{\infty} 1/(2^i + 1) \approx 1.264$ is schedulable. This significantly improves upon the current best-known density bound of 1.546 and is, in fact, optimal.

2 Introduction. Scheduling problems are a broad class of problems in computer science that involve assigning jobs to machines with specific time constraints. Applications range from industrial optimization to the entertainment industry to the functioning of operating systems [1].

The pinwheel problem is a well-studied periodic scheduling problem in which there is a single machine and a finite number of jobs that reappear immediately after being processed. For example, a vending machine may need to be restocked at least once every week, with the maintenance deadline depending on the last time it was restocked. Formalizing this, a pinwheel scheduling instance is a list of integers $(a_1, a_2, a_3, \cdots, a_n)$. The pinwheel decision problem asks whether a pinwheel scheduling instance has a valid schedule, in particular, a map $f : \mathbb{N} \to [n]$

*Department of Computer Science, Cornell University, Ithaca, NY (abm247@cornell.edu).

corresponding to an assignment of days to jobs 1 through n such that for all $i \in [n]$, every sequence of a_i consecutive days has job i assigned at least once.

The discretized point patrolling problem asks, for an instance $(a_1, a_2, a_3, \ldots, a_n)$, whether a valid schedule exists, in particular, a map $f : \mathbb{N} \to [n]$ corresponding to an assignment of days to jobs such that for all $i \in [n]$, every sequence of a_i consecutive days has job i assigned at *most* once. We also refer to this as the pinwheel covering problem, distinguishing it from the "traditional" pinwheel problem by qualifying the latter as the pinwheel packing problem for the remainder of this paper.

Consider the fraction of all the days each job takes up for a pinwheel packing instance. It is at least $\frac{1}{a_i}$ for job i. We can formulate a notion of density of an instance as the sum of these fractions (i.e., $\sum_{i=1}^{n} \frac{1}{a_i}$), and we can clearly see that if the density is greater than 1, no valid pinwheel schedule exists. However, density being less than or equal to 1 is not sufficient for schedulability. For example, the instance $(2, 3, 6)$ has density $\frac{1}{2} + \frac{1}{3} + \frac{1}{6} = 1$ but there is no valid schedule. Intuitively, the tasks with periods two and the three take up all the spaces, leaving no room for the task with period six. This example can be generalized.

Fact 1. $(2, 3, a_3)$ *is not schedulable for any* $a_3 \in \mathbb{N}$ *[14].*

Fact 1 is proven in Appendix A.1, and applying the fact, we know that for all $k \in \mathbb{N}$, there exists a pinwheel scheduling instance that is not schedulable with density $\frac{1}{2} + \frac{1}{3} + \frac{1}{i} = \frac{5}{6} + \frac{1}{i}$.

For pinwheel covering, we again define $D(A) = \sum_{a_i \in A} \frac{1}{a_i}$, where now if the density is less than 1, an instance cannot be schedulable. There is a corresponding version of Fact 1 in the covering setting.

Fact 2. $(2, 3, 5, 9, \cdots, 2^k + 1)$ *is not schedulable for any* $k \in \mathbb{N}$ *(Theorem 17 of [17]).*

Fact 2 is proven in subsection A.2, and applying the fact, we know that for all $i \in \mathbb{N}$, there exists a pinwheel covering instance that is not schedulable with density

$$\sum_{n=0}^{k} \frac{1}{2^n + 1} = \sum_{n=0}^{\infty} \frac{1}{2^n + 1} - \sum_{n=k+1}^{\infty} \frac{1}{2^n + 1} \geq - \sum_{n=k+1}^{\infty} \frac{1}{2^n} + \sum_{n=0}^{\infty} \frac{1}{2^n + 1} = -\frac{1}{2^k} + \sum_{n=0}^{\infty} \frac{1}{2^n + 1}$$

In this paper, the phrase "density bound" refers to a density threshold that provides a schedulability guarantee. For the packing setting, a density bound of d_p implies that every instance with density at most d_p is schedulable. In the covering setting, a density bound of d_c implies that every instance with density at least d_c is schedulable.

2.1 Related Work. Fact 1 shows that a density bound of $\frac{5}{6}$ for the pinwheel packing problem would be optimal, and this was conjectured to be possible. There was significant work towards this goal, with bounds of $\frac{1}{2}$ [13], $\frac{2}{3}$ [4], $\frac{7}{10}$ [3], and $\frac{3}{4}$ [7]. Finally, Kawamura [14] proved the conjectured $\frac{5}{6}$ bound. One particularly relevant technique used in the $\frac{5}{6}$ proof is the concept of a fold operation, which enables us to reduce the problem to a finite number of instances and delegate the analysis to a computer [10]. We repurpose the folding idea (Algorithm 1) to the covering setting and use the same folding operation of Kawamura [14] for bamboo garden trimming.

Note that we have used $A \sqcup (a)$ as the notation for adding a job of period a to the instance A, and $A \ominus (a)$ for removing a job of period a (with multiple periods in parenthesis indicating multiple

Algorithm 1: Fold Operation $\mathrm{pfold}_\theta(A)$.

Input: Pinwheel instance $A = (a_1, a_2, \ldots, a_n)$
while $\max(A) > \theta$ **do**
  Let a and b be the largest and second largest values in A, allowing for repetition.
  if $b > \theta$ **then**
    $A \leftarrow A \ominus (a, b)$
    $A \leftarrow A \sqcup (b/2)$
  else
    $A \leftarrow A \ominus (a)$
    $A \leftarrow A \sqcup (\theta)$
  return A

additions or removals). The usefulness of Algorithm 1 relates to two properties of pinwheel packing instances, which can be used to show that pfold_θ preserves instance unschedulability.

LEMMA 2.1. *If $A \sqcup (a)$ is schedulable, then $A \sqcup (b)$ is schedulable for any $b \geq a$ (Lemma 3 of [14]).*

LEMMA 2.2. *If $A \sqcup (a)$ is schedulable, then $A \sqcup (\underbrace{a \cdot q, a \cdot q, \ldots, a \cdot q}_{q})$ is schedulable (Lemma 3 of [14]).*

LEMMA 2.3. *For all pinwheel packing instance A, if A is unschedulable, then for all θ, $pfold_\theta(A)$ is unschedulable (Lemma 4 of [14]).*

Lemma 2.1 is called the monotonicity property, and Lemma 2.2 is called the partitioning property.

Just as $\frac{5}{6}$ is an optimal density bound for the packing setting, in the covering setting, due to Fact 2, if a density bound of $\sum_{i=0}^{\infty} \frac{1}{2^i+1}$ were true, it would be optimal. We refer to the conjecture that this density bound is indeed possible (i.e., every instance with density at least $\sum_{i=0}^{\infty} \frac{1}{2^i+1}$ is schedulable) as the $2^i + 1$ conjecture, and it was originally posed by Kawamura and Soejima (Conjecture 18 of [17]). This conjecture is particularly relevant due to the relatively recent resolution of the $\frac{5}{6}$ conjecture [14] after over 30 years. Currently, the best-known density bound is 1.546 [17].

2.2 Contributions. This paper slice presents improves the state-of-the-art density bound for discretized point patrolling from 1.546 [17] to the optimal value of $\sum_{i=0}^{\infty} \frac{1}{2^i+1} \approx 1.264$. To do so, this paper extends the notions of monotonicity and partitioning to pinwheel covering and introduces two fold operations for pinwheel covering instances, which may be of independent interest.

Several of our claims involve search over a finite number of cases, similar to that found in [14]. Supporting computational evidence for these claims is publicly available at: https://github.com/ahanbm/optimal-dpp

An independent and concurrent work by Kawamura and Kobayashi [15] matches our $\sum_{i=0}^{\infty} \frac{1}{2^i+1}$ density bound for pinwheel covering.

3 Technical Overview.

In this section, we provide the high-level ideas and algorithms for our results in pinwheel covering and bamboo garden trimming.

3.1 Pinwheel Covering.

We start by setting up the basics for pinwheel covering.

We define a fractional pinwheel covering problem, analogous to the fractional packing problem:

Consider a fractional instance $(a_1, a_2, a_3, \ldots, a_n)$ with $a_i \in \mathbb{R}^+$ for all $i \in [n]$. We define a valid schedule as one in which days are assigned to jobs in a way so that for all $i \in [n]$ and for all $d \in \mathbb{N}$, every sequence of d consecutive days has job i scheduled at most $\lceil \frac{r}{a_i} \rceil$ times. We now formulate the analogue of the monotonicity and partitioning properties [14] in the covering setting.

LEMMA 3.1. *Monotonicity (Covering): If $A \sqcup (a)$ is schedulable, then $A \sqcup (b)$ is schedulable for any $b \leq a$.*

Proof. Given a valid schedule for $A \sqcup (a)$, we can reassign the days where the job with period a is scheduled to the job with period b. ∎

LEMMA 3.2. *Partitioning (Covering): If $A \sqcup (a)$ is schedulable, then $A \sqcup (\underbrace{a \cdot q, a \cdot q, \ldots, a \cdot q}_{q \text{ times}})$ is schedulable.*

Proof. Given a valid schedule for $A \sqcup (a)$, we can sequentially schedule the q jobs of period $a \cdot q$ within the slots assigned to the job of period a, returning to the first job of period $a \cdot q$ after all are scheduled and repeating. Since there is at least a gap of a between each scheduling of the original job of period a, there is a gap of at least $a \cdot q$ between repetitions of the same job of period $a \cdot q$ in the new instance. ∎

We now define a fold operation $\text{cfold}_\theta(A)$, which is intended to take a pinwheel covering instance A and transform it into an instance A' such that no element of A' is greater than $\theta \in \mathbb{N}$, and such that unschedulability is preserved. Algorithm 2 shows the major steps.

Algorithm 2: Fold Operation $\text{cfold}_\theta(A)$

Input: Pinwheel covering instance $A = (a_1, a_2, a_3, \ldots, a_n)$
while the maximum value $a \in A$ satisfies $a > \theta$ **do**
  $A \leftarrow A \ominus (a)$
  $b \leftarrow \max(A)$
  $A \leftarrow A \ominus (b)$
  if $a < 2b$ **then**
    $A \leftarrow A \sqcup (\frac{a}{2})$
  else
    $A \leftarrow A \sqcup (b)$
return A

As in Algorithm 1, we use $\ominus$ to represent removing a job and $\sqcup$ to represent adding one. Algorithm 2 can be regarded as the natural analogue of the fold operation for the packing setting introduced by Kawamura [14]. It can be shown that Algorithm 2 preserves unschedulability and decreases the density of A by at most $\frac{2}{\theta}$. Similar to the fold operation in [14], Algorithm 2 can be leveraged to prove a density bound of 1.3 for discretized point patrolling. Since Algorithm 2 preserves schedulability, we only need to consider instances with elements at most θ whose density is at least $1.3 - \frac{2}{\theta}$. Due to the fractional setting, the set of all possible instances with elements at most θ is infinite. We distill a finite set of instances by taking the ceiling of each element, and then leverage use computer analysis. A density bound of 1.3 is already a substantial improvement over the current state-of-the-art of 1.546, but proving the $2^i + 1$ conjecture requires more tools.

We briefly stop to justify why the current fold operation is insufficient. The density loss for our fold operation is $\frac{2}{\theta}$ while the density loss for Kawamura's fold operation is $\frac{1}{\theta}$, so we might hope that the parameter 2 can be improved. However, it is tight, e.g., consider the instance $[\theta + 1, 2\theta + 1, 4\theta + 1, \ldots, 2^n \cdot \theta + 1]$ for where n is large and θ is a large power of 2. In order for the method described to work, we need to be able to schedule all instances of elements at most θ with density at least $-\frac{2}{\theta} + \sum_{i=0}^{\infty} \frac{1}{2^i+1}$. However, the instance $[2, 3, 5, \ldots, 2^{\lceil \log_2 \theta \rceil - 1} + 1]$ is within the desired density bound, and we have already shown it is not schedulable in Fact 2.

Fact 3. *Let $A(\theta) = [2, 3, 5, \ldots, 2^{\lceil \log_2 \theta \rceil - 1} + 1]$. For any θ, $D(A(\theta)) \geq -\frac{2}{\theta} + \sum_{i=0}^{\infty} \frac{1}{2^i+1}$.*

Fact 3 is proved in subsection A.3. This reasoning demonstrates that Algorithm 1 is insufficient to prove an optimal bound directly, so we develop an improved fold algorithm.

Algorithm 3: Improved Fold Operation $\mathrm{cfold}_\theta^{\mathrm{imp}}(A)$

Input: Pinwheel covering instance $A = (a_1, a_2, a_3, \ldots, a_n)$ and $\theta = 2^i$ for $i \in \mathbb{Z}_{>1}$
$A_{\max} \leftarrow \max(A)$
$\theta_{\mathrm{current}} \leftarrow 2^{\lceil \log_2(A_{\max}) \rceil - 1}$
while $\theta_{current} \geq \theta$ **do**
  Let n be the number of elements of A in the range $(\theta_{\mathrm{current}}, 2\theta_{\mathrm{current}}]$
  if n is odd and $n \geq 3$ **then**
    Let $m_1 \leq m_2 \leq m_3$ be the three smallest elements of A in the range $(\theta_{\mathrm{current}}, 2\theta_{\mathrm{current}}]$
    if $m_3 \leq \frac{4}{3}\theta_{current}$ **then**
      $A \leftarrow A \ominus (m_1, m_2, m_3)$
      $A \leftarrow A \sqcup \left(\frac{m_3}{3}\right)$
  $A \leftarrow \mathrm{cfold}_{\theta_{\mathrm{current}}}(A)$
  $\theta_{\mathrm{current}} \leftarrow \theta_{\mathrm{current}}/2$
return A

Algorithm 3 achieves a better guarantee than a density loss of $\frac{2}{\theta}$ in specific circumstances. In particular, it can be shown that for a single iteration, if there are at least two elements between 2^{i-1} and 2^i, the algorithm achieves a density loss bound of $\frac{3}{4\theta}$. This bound is in contrast to the one iteration density loss bound for cfold_θ of $\frac{1}{\theta}$. In addition, the improvement in density loss can be iterated under circumstances that we establish.

This improvement enables us, with computer search, to determine that all pinwheel covering instances with density at least $\sum_{i=0}^{\infty} \frac{1}{2^i+1}$ are schedulable. The computer analysis is similar to that used for the 1.3 density bound but with more detailed casework.

4 An Improved Density Bound for Pinwheel Covering.

In this section, we prove a 1.3 density bound for pinwheel covering. We rely on Algorithm 2 along with a new program for pinwheel covering adapting some computer search ideas from the packing setting [10]. We start by proving properties of Algorithm 2.

LEMMA 4.1. *Every element of $A' = cfold_\theta(A)$ is at most θ.*

Proof. We can clearly see from the while loop condition in Algorithm 2 that upon termination, the maximum value of the returned A must be no greater than θ. Furthermore, the algorithm terminates because, at every iteration of the while loop, the length of A decreases by 1; thus, there are at most n iterations. ∎

LEMMA 4.2. *If A is unschedulable, then $cfold_\theta(A)$ is unschedulable.*

Proof. We prove the contrapositive: if $\text{cfold}_\theta(A)$ is schedulable, then A is schedulable.

We prove the claim by induction on the number of remaining while loop iterations. For the base case, when there are zero iterations left, the returned value is assumed to be schedulable. If the instance A_r, the value of A with r remaining while loop iterations, is schedulable, then we need to prove A_{r+1} is also schedulable. There are two cases: either the a and b (highest and second-highest values) in A_{r+1} were replaced by $\left(\frac{a}{2}\right)$ (in the case $a < 2b$) or by (b). In the case of (b), A_{r+1} is the same as A_r but with an extra job, so A_{r+1} is schedulable according to the monotonicity property (we can treat A_r as having an ∞ period job).

In the case where $a < 2b$ and the $\left(\frac{a}{2}\right)$ job in A_r arose from the a and b jobs in A_{r+1}, we know that schedulability of $(A_r \ominus \left(\frac{a}{2}\right)) \sqcup \left(\frac{a}{2}\right)$ implies schedulability of $(A_r \ominus \left(\frac{a}{2}\right)) \sqcup (a, a)$ (by the partitioning property) and this in turn implies schedulability of $(A_r \ominus \left(\frac{a}{2}\right)) \sqcup (a, b) = A_{r+1}$ (by the montonicity property). Since we covered both possible cases, this completes the inductive step. Inducting till $A_{n'}$ where n' is the total number of while loop iterations to evaluate $\text{cfold}_\theta(A)$, we have that A is schedulable, as desired. ∎

LEMMA 4.3. $D(\text{cfold}_\theta(A)) \geq D(A) - \frac{2}{\theta}$

Proof. We split the density loss analysis into two parts: the last while loop iteration and the rest. For any iteration, we claim that the density loss is at most $\frac{1}{b} - \frac{1}{a}$, where a and b are the maximum and second-maximum values before that iteration. In particular, if a is deleted, then $a \geq 2b$ so $b \leq \frac{a}{2}$ and the density loss is $\frac{1}{a} = \frac{2}{a} - \frac{1}{a} \leq \frac{1}{b} - \frac{1}{a}$. If (a, b) is replaced with $\left(\frac{a}{2}\right)$ then the density loss is $\frac{1}{a} + \frac{1}{b} - \frac{2}{a} = \frac{1}{b} - \frac{1}{a}$, with direct equality. The total density loss for all iterations but the last is

$$\frac{1}{b_1} - \frac{1}{a_1} + \frac{1}{b_2} - \frac{1}{a_2} + \frac{1}{b_3} - \frac{1}{a_3} + \cdots + \frac{1}{b_{x-1}} - \frac{1}{a_{x-1}}$$

where we have said that a_i and b_i are the maximum and second-maximum values before the ith iteration, and there are x total iterations. Furthermore, $b_x \geq a_{x+1}$ for all x, since at each iteration a either gets deleted or replaced with $\frac{a}{2}$ in the case that $\frac{a}{2} < b$. Therefore, the sum above can be upper-bounded by a telescoping sum:

$$\frac{1}{b_{x-1}} - \frac{1}{a_{x-1}} + \frac{1}{a_{x-1}} - \frac{1}{a_{x-2}} + \cdots + \frac{1}{a_2} - \frac{1}{a_1} = \frac{1}{b_{x-1}} - \frac{1}{a_1} \leq \frac{1}{b_{x-1}}$$

Since there is another iteration and $a_x \leq b_{x-1}$, we must have that $b_{x-1} \geq \theta$ so the total density loss in all iterations but the last is at most $1/\theta$. For the last iteration, we can make a similar claim as before and say that the density loss is at most $1/a$ at any iteration, by similar logic as before. Either a is deleted for exact loss of $\frac{1}{a}$ or the loss is $\frac{1}{a} + \frac{1}{b} - \frac{2}{a} = \frac{1}{b} - \frac{1}{a}$ and this is in the case $a < 2b$ so $\frac{1}{b} - \frac{1}{a} < \frac{2}{a} - \frac{1}{a} = \frac{1}{a}$. Therefore, the loss in the last iteration is at most $1/\theta$, for a total density loss bound of

$$\frac{1}{\theta} + \frac{1}{\theta} = \frac{2}{\theta}$$

as desired. ∎

LEMMA 4.4. *Any instance $A = (a_1, a_2, \ldots, a_n)$ whose periods are integers at most 16 and satisfies $D'(A) \geq 1.3 - \frac{2}{16}$ is schedulable, where*

$$D'(A) = \sum_{i=1}^{n} \begin{cases} 1/a_i & \text{if } a_i \leq 8, \\ 1/(a_i - 1) & \text{if } a_1 > 8. \end{cases}$$

Proof. The set of instances of such A is infinite, but we can consider the restricted subset R that contains only those A that would be underneath the density bound if another element were to be removed. Specifically, to convert a general A into an element of R, we remove the maximum element of A until another removal would cause $A_{removed}$ to go under the density bound of $1.3 - \frac{2}{16}$. By monotonicity, if every element of R is schedulable, then Lemma 4.4 is true. Since R contains a finite number of elements, it suffices to compute a schedule for each possibility, which we have done. ∎

THEOREM 4.5. *Every pinwheel covering instance with density at least 1.3 is schedulable.*

Proof. Suppose, for the sake of contradiction, that there exists an unschedulable instance A with $D(A) \geq 1.3$. Then, $\text{cfold}_{16}(A)$ is also unschedulable, according to Lemma 4.2. In addition, $D(\text{cfold}_{16}(A)) \geq D(A) - \frac{2}{16}$. Consider A' which contains a for all $a \in \text{cfold}_{16}(A)$ such that $a \leq 8$, and additionally contains $\lceil a \rceil$ for all elements $a \in \text{cfold}_{16}(A)$ such that $a > 8$, with appropriate multiplicity. Since $\text{cfold}_{16}(A)$ is unschedulable, A' is also unschedulable according to the monotonicity property. In addition, since the elements of $\text{cfold}_{16}(A)$ that are at most 8 are unchanged, and the rest increase by at most 1, $D'(A') \geq D(\text{cfold}_{16}(A))$, implying that $D'(A') \geq D(A) - \frac{2}{16}$ and since $D(A) \geq 1.3$, $D'(A') \geq 1.3 - \frac{2}{16}$. Furthermore, because $\text{cfold}_{16}(A)$ produces elements that are at most 16 according to Lemma 4.1, A' does not increase elements above 16, and every element of A' is an integer, Lemma 4.4 applies. In particular, A' is schedulable, a contradiction. This proves a bound of 1.3 for pinwheel covering. ∎

Acknowledgments. I would like to gratefully acknowledge Prof. Robert Kleinberg for introducing me to the pinwheel problem and for his guidance. I would also like to acknowledge the helpful suggestions of three anonymous reviewers.