Online Demand Strip Packing

Sebastian Bruchhold
Institut für Mathematik, Technische Universität Berlin, Germany

Franziska Eberle
Institut für Mathematik, Technische Universität Berlin, Germany

Georgios Moneftsis
Institut für Mathematik, Technische Universität Berlin, Germany

Malin Rau
Department of Mathematical Sciences, Chalmers University of Technology & University of Gothenburg, Sweden

Albert Vesterlund
Department of Mathematical Sciences, Chalmers University of Technology & University of Gothenburg, Sweden

Abstract

In the Demand Strip Packing problem (DSP), we are given a finite set of tasks, each characterized by a specific duration and energy demand. These tasks need to be scheduled non-preemptively within a given time frame while minimizing the peak demand: the maximum amount of energy consumed by the tasks being executed at any point in time. We are the first to consider the *online* variant of the problem, where tasks are revealed to an algorithm one by one in a list. Upon arrival, each task must be assigned an irrevocable starting time before the next task in the list is revealed. As usual in online optimization, we evaluate the performance of online algorithms using competitive analysis. We give a strictly 4.263-competitive algorithm for Online DSP, which is stronger than the respective bound of 6.479 for the related problem Online Strip Packing. Additionally, we prove a lower bound of 1.812 on the competitive ratio of any online algorithm for DSP and, thus, clearly separate Online DSP from Online Minimum Peak Appointment Scheduling (MPAS), a special case of Online DSP, for which a strictly $\frac{5}{3}$-competitive algorithm is known.

2012 ACM Subject Classification Theory of computation → Online algorithms; Theory of computation → Scheduling algorithms; Theory of computation → Packing and covering problems

Keywords and phrases Online Demand Strip Packing, competitive analysis, scheduling, packing

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.54

Supplementary Material *Software (Source Code)*: `https://gitlab.com/bruchhold/online-demand-strip-packing`, archived at `swh:1:dir:7b427dffa87e26a822ae9e0bdaa9dcdcf2854ce8`

Funding The first and second author are funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – The Berlin Mathematics Research Center MATH+ (EXC-2046/1, EXC-2046/2, project ID: 390685689).
Malin Rau: Supported by the Swedish Research Council under grant number 2025-04724.

1 Introduction

We investigate a novel online scheduling problem, Online Demand Strip Packing. Even though its offline counterpart has received considerable attention over the last five years [8, 9, 13, 17], and special cases [10, 27] as well as related problems [16, 32, 33] have been studied in the online setting as well, we are the first to consider the online variant of the problem.

In the Demand Strip Packing problem (DSP), we are given a *time frame* $[0, T]$ for some $T > 0$ and a set $\mathcal{J}$ of jobs, where each job j has a fixed duration (or processing time) p_j as well as an energy demand e_j and needs to be scheduled in $[0, T]$, i.e., for each job in $\mathcal{J}$, we must decide on a *starting time* $0 \leq s_j \leq T - p_j$. At any point in time $t \in [0, T]$, we define the *load* at t as the total energy demand of all jobs that are running at time t, i.e., $\text{load}(t) \coloneqq \sum_{j : s_j \leq t < s_j + p_j} e_j$, and the objective is to *minimize the maximum total energy demand* or maximum load over the time frame.

In Online Demand Strip Packing, the jobs in $\mathcal{J}$ are revealed one after the other in a list to the algorithm, which is forced to irrevocably assign a starting time to a just-revealed job before observing the next job. We typically overload notation and let $\mathcal{J}$ be both the set of jobs as well as the tuple of jobs in the order in which they are revealed online.

An important application of DSP arises in the context of smart-grid scheduling: The *smart grid* presents a new paradigm for the production and distribution of electricity, which offers potential solutions to the challenges posed by the integration of renewable energy sources. A critical novelty of the smart grid is the bidirectional flow of information between energy producers and consumers [23]. In particular, smart-grid technologies allow grid operators to control not only the supply side but also the demand side and to adapt industrial (e.g., steel production) and household processes (e.g., charging an electric vehicle) according to energy production patterns, which are heavily weather dependent and, as such, highly volatile for most renewable energy sources, such as wind and solar. This stands in stark contrast to fossil-fuel-driven electricity production, which can more readily meet the increasing demand due to population growth, economic activity, and technological progress, but causes significant harm to the environment.

When modeling each industrial and household process as an individual job with duration and energy demand, DSP allows for minimizing the maximum energy demand over a given time frame, i.e., over a day or a week. We believe that developing good algorithms for DSP is an important stepping stone in matching energy consumption patterns to its production patterns as eventually required in smart-grid scheduling.

DSP has recently been the subject of extensive research [8, 9, 13, 17]. Because the problem is known to be NP-hard, research has focused on polynomial-time and pseudo-polynomial approximation algorithms[1]. In fact, the question of approximating DSP in polynomial time is effectively settled, since a recently developed polynomial-time algorithm attains an approximation ratio of $\frac{3}{2} + \varepsilon$ [9], while it is known that no polynomial-time algorithm can have an approximation ratio below $\frac{3}{2}$ unless $P = NP$ [30].

While these algorithms are major theoretical results, they are based on the key assumption that the scheduler possesses all information about the instance at hand. Especially in a real-life grid, this does not have to be the case since energy-consuming processes are inherently uncertain. A smart-grid system may be required to assign starting times while only being aware of a small fraction of the jobs to be scheduled. As a result and due to its theoretical importance, Online Demand Strip Packing is of interest, and we are the first to investigate it.

As usual, we evaluate the performance of online algorithms using competitive analysis; that is, we determine the worst-case ratio between the maximum energy load $\text{Alg}(\mathcal{J})$ achieved by an online algorithm Alg and the maximum energy load $\text{Opt}(\mathcal{J})$ of an optimal offline solution over all possible instances $\mathcal{J}$. More precisely, we say that an algorithm is *ϱ-competitive* if there is a constant $\kappa \geq 0$ such that, for all instances $\mathcal{J}$, we have

$$\text{Alg}(\mathcal{J}) \leq \varrho \cdot \text{Opt}(\mathcal{J}) + \kappa.$$

An algorithm is *strictly ϱ-competitive* if $\kappa = 0$ holds in the above definition. The infimum ϱ_{Alg} such that Alg is strictly ϱ_{Alg}-competitive is called the *competitive ratio*.

[1] An *α-approximation algorithm* is a polynomial-time algorithm whose maximum load is within a factor α of that of an optimal schedule.

While related problems such as Minimum Peak Appointment Scheduling (MPAS) and Strip Packing have been considered in an online setting, we are the first to investigate Online DSP, and we hope to foster future research. We design a strictly 4.263-competitive algorithm for Online DSP and, hence, distinguish Online DSP from Online Strip Packing, for which the current best upper bound is 6.479. We complement this result by giving a lower bound of 1.812 on the competitive ratio of any deterministic algorithm and, thus, separate Online DSP from Online MPAS, for which a strictly $\frac{5}{3}$-competitive algorithm is known.

Further Related Problems

In this section, we give an overview of several online optimization problems that are related to Online DSP and discuss respective results. The *asymptotic competitive ratio* is also used to determine the asymptotic behavior of an algorithm for instances where $\textsc{Opt}(\mathcal{J})$ is large, e.g., significantly larger than the maximum energy demand $\max_{j\in\mathcal{J}} e_j$, and it is defined as $\inf\left\{\varrho \,\middle|\, \exists\kappa\colon \textsc{Alg}(\mathcal{J}) \le \varrho\cdot\textsc{Opt}(\mathcal{J})+\kappa \text{ for all } \mathcal{J} \text{ with } \max_{j\in\mathcal{J}} e_j \le 1\right\}$.

Online Bin Packing

In the Online Bin Packing problem, items of width in $[0,1]$ and constant height appear one by one, and any algorithm must pack them into bins of width 1. The objective of any algorithm is to minimize the number of bins used. The competitive ratio for Online Bin Packing is $\frac{5}{3}$ due to [3]. After a series of algorithms for Online Bin Packing [2, 18, 19, 21, 26], the current best algorithm (in terms of the asymptotic competitive ratio) is the 1.579-competitive Advanced Harmonic algorithm by [2]. After a long line of research [22, 28, 29], the current best lower bound on the asymptotic competitive ratio is 1.542 by [5].

While Bin Packing algorithms cannot directly be applied to Online (Demand) Strip Packing, partitioning the jobs according to their energy demands enables us to use Bin Packing algorithms *for each height group* individually, similarly to a technique for Online Strip Packing, which we discuss next.

Online Strip Packing

In this problem, a sequence of rectangular jobs is given to the algorithm, which must be placed without rotation into a strip of width T and infinite height, such that the interiors of the rectangles are pairwise disjoint. In other words, the algorithm must assign to each job *both* a starting time and a starting height such that no two jobs overlap. The objective is to minimize the maximum height of the resulting packing. We refer to any feasible solution as a *feasible geometric packing*. While any feasible geometric packing corresponds to a feasible schedule of at most the same height, i.e., maximum energy load, the reverse is not true [7, 13]. In fact, there are instances where the two optima are a factor $\frac{5}{4}$ apart [13].

A class of algorithms that have been intensely studied in the context of Online Strip Packing are *shelf algorithms*: They create horizontal bins (*shelves*) in the strip for jobs of similar height. Two jobs belong to the same class if both their heights are in $(r^{k+1}, r^k]$ for some $k\in\mathbb{Z}$; shelves for that class have a height of r^k. For each class, incoming jobs are placed into a shelf associated with that class using an Online Bin Packing algorithm, creating a new shelf on top of the current packing whenever the Bin Packing algorithm opens a new bin. To construct a shelf algorithm, it thus suffices to decide on an Online Bin Packing algorithm and a value of the parameter $r\in(0,1)$. The current best algorithms are indeed generalizations of shelf algorithms: The algorithm by Yu, Mao, and Xiao [33] achieves a competitive ratio of 6.479, while the current best lower bound of 2.618 by Yu, Mao, and Xiao [32] still leaves a wide gap. The asymptotic competitive ratio is much better understood: The algorithm by Han et al. [15] achieves an asymptotic competitive ratio of 1.589, and the lower bound of 1.542 by Balogh et al. [5] almost matches it.

Online Minimum Peak Appointment Scheduling

The important special case of Online DSP with uniform energy demands, i.e., $e_j = 1$ for all $j \in \mathcal{J}$, is known as Online Minimum Peak Appointment Scheduling (MPAS), sometimes also as Online Minimum Peak Job Scheduling (MPJS). We are again interested in minimizing the maximum load at any point in $[0, T]$, which is equivalent to the number of jobs that are active at that point. While this problem can also be solved using an Online Bin Packing algorithm, the key difference is that an algorithm for MPAS only has to assign each job a starting time but *not* a bin into which the job should be placed. Thus, a natural question is whether a schedule given by an algorithm for MPAS can be translated into a feasible Bin Packing solution with max-load many bins (in hindsight).

Escribe, Hu, and Levi [10] answer this question in the affirmative giving a graph-theoretic proof; Moneftsis [24] also gives a combinatorial proof. Hence, the offline optimal solutions of MPAS and Bin Packing coincide, implying that *upper* bounds on the competitive ratio immediately carry over. Therefore, there is a strictly $\frac{5}{3}$-competitive algorithm for Online MPAS [3], and 1.579 also is an upper bound on the asymptotic competitive ratio [2]. It is currently not known whether the higher flexibility in Online MPAS allows for improved bounds on the deterministic competitive ratios; it is true for randomized algorithms [10, 27], which we do not detail here. Bearing in mind that lower bounds from Online Bin Packing do not necessarily hold for Online MPAS, the currently best lower bound on the competitive ratio is 1.5 due to Escribe, Hu, and Levi [10] and on the asymptotic competitive ratio is 1.269 due to Balogh et al. [4]. We are able to clearly separate Online MPAS from Online DSP in terms of the competitive ratio by giving a lower bound of 1.812, which is above the best such guarantee for Online MPAS.

Online Makespan Minimization

The special case of Online DSP with uniform processing times, i.e., $p_j = 1$ for $j \in \mathcal{J}$ and $T \in \mathbb{N}$, can be shown to be equivalent to Online Makespan Minimization by identifying the processing time with the machine requirement, and the energy demand with the processing requirement; a job starting at time $t \in \mathbb{N}_0$ is interpreted as being assigned to machine $i = t+1$ and $m = T$ is the number of machines. We are interested in assigning each job j to a machine $i \in [m]$, which increases its load by the processing requirement of j, such that the maximum load is minimized. After a long line of research [1, 6, 11, 12, 14, 20], starting with Graham's $\left(2 - \frac{1}{m}\right)$-approximate List Scheduling algorithm [14], the current best algorithm by Fleischer and Wahl achieves a competitive ratio of 1.92 [11], while the best lower bound of 1.88 is due to Rudin [25]. We conjecture that Online Makespan Minimization admits better algorithms than Online DSP.[2]

[2] Unfortunately, we realized the connection between the two problems only after submission. We would like to point out that the known lower bound of 1.88 for Online Makespan Minimization subsumes our lower bound of 1.812 for Online DSP.

2 The Alternating and Closing First Fit Algorithm

We present our shelf-inspired algorithm ALTERNATING AND CLOSING FIRST FIT$_r$, ACFF$_r$, where r is the shelf-height parameter. We complement its upper bound of 4.263 by providing a lower bound of 2.910 on the competitive ratio of ACFF$_r$ for $r \in (0, 1)$.

2.1 The Algorithm

Given a list $\mathcal{J}$ and a time horizon T, a job $j \in \mathcal{J}$ is *wide* if $p_j > \frac{T}{2}$ and *narrow* otherwise. We denote the respective subsets of jobs by $\mathcal{W}$ and $\mathcal{N}$. The ACFF$_r$ algorithm relies on a parameter $r \in (0, 1)$ that is used to geometrically partition jobs based on their energy demand. Borrowing notation from Online Strip Packing, we assign each job a *shelf height* r^k such that $e_j \in (r^{k+1}, r^k]$ and then pack jobs into their respective shelves using the FIRST FIT algorithm (FF) with three important twists:

- We only schedule *narrow* jobs with FF (similar to [16, 33] for Online Strip Packing).
- We *close* a shelf and no longer use it to schedule further jobs if a certain condition is met.
- We use *left-* and *right-justified* shelves.

More concretely, upon receiving a wide job j, we set its starting time to 0 and do not assign it to any shelf. Conversely, a narrow job j is scheduled in a shelf of height r^k with $e_j \in (r^{k+1}, r^k]$ according to standard FF. That is, we assign the job to the lowest, i.e., earliest created, shelf of height r^k where it still fits and, if no such shelf exists, we open a new shelf.

For $k \in \mathbb{Z}$ odd, all shelves of height r^k are left-justified, i.e., a new job assigned to such a shelf either starts at 0 (if it is the first such job) or at the end time of the job previously assigned to the shelf if the total width of jobs in the shelf does not exceed T. For $k \in \mathbb{Z}$ even, all shelves of height r^k are right-justified, i.e., a new job either finishes at T or at the starting time of the previous job.

After placing a second job into a shelf, we close the shelf, i.e., we place no further jobs in it, if the combined processing time of the two (narrow) jobs is greater than $\left(1 - \frac{r}{2}\right)T$.

2.2 The Analysis

We show the following upper bound on the competitive ratio of ALTERNATING AND CLOSING FIRST FIT$_r$ with left- and right-justified shelves for Online Demand Strip Packing.

▶ **Theorem 1.** *There is a parameter $\varphi \in \left[\frac{2}{3}, 1\right]$ such that, for any constant $r \in [\varphi, 1)$ and any job list $\mathcal{J}$,* ALTERNATING AND CLOSING FIRST FIT$_r$ *achieves maximum energy demand*

$$\text{ACFF}_r(\mathcal{J}) \leq \left(\max\left\{\frac{4}{3r}, 2\right\} + 1 + \frac{r}{1-r^2}\right) \text{OPT}(\mathcal{J}) = \left(3 + \frac{r}{1-r^2}\right) \text{OPT}(\mathcal{J}).$$

For $r = \varphi$, algorithm ACFF$_r$ *achieves a competitive ratio of approximately* 4.263.

Even though our algorithm ACFF$_r$ does not behave fully symmetrically (with respect to $\left[0, \frac{T}{2}\right]$ and $\left[\frac{T}{2}, T\right]$ due to the left- and right-justification of shelves), we will focus on having peak demand occur at $t \in \left[0, \frac{T}{2}\right]$ during the analysis and take care that all arguments generalize easily to $t \in \left[\frac{T}{2}, T\right]$ as well.

Overview of the Analysis. The crucial ingredient that distinguishes the analysis of shelf algorithms for Online DSP from those for Online Strip Packing and allows us to go beyond the current limitations of the analysis is the following: While, in Strip Packing, the heights of shelves cannot be adapted retrospectively, in DSP, we care about the *actual load* contributed by the jobs in $\mathcal{J}$ at any given point $t \in [0, T]$; sometimes we are interested in the load from a subset of jobs $\mathcal{J}' \subseteq \mathcal{J}$ and we denote this by $\text{load}_{\mathcal{J}'}(t)$. Often, we in fact employ upper bounds on $\text{load}_{\mathcal{J}'}(t)$ such as the total height of the shelves to which the jobs in $\mathcal{J}'$ were assigned. (This is the *only* feasible bound in Strip Packing.) A more fine-grained upper bound is the total effective height of all *shelves* with jobs in $\mathcal{J}'$, where the *effective height* of a shelf is the maximum energy demand of any job assigned to the shelf. When bounding the load at a point t, we distinguish the energy demand required by wide jobs, closed shelves, dense shelves, and sparse shelves. A shelf that was not closed, of a given height r^k, is called *sparse* if it is the last non-closed shelf of height r^k created; otherwise it is *dense*. Intuitively, we can bound the total load contributed by wide jobs and by closed or dense shelves in terms of the area of the respective jobs. For wide jobs, the total load is at most $\frac{2}{T}$ times their respective area (cf. Observation 3). For jobs in closed shelves, their total load is also at most $\frac{2}{T}$ times their respective area (cf. Proposition 4). Informally, for almost all dense shelves, the ratio between the total height of these shelves and the area of the respective jobs is at most $\frac{4}{3rT}$ (cf. Lemma 6); the proof crucially relies on the closing rule of ACFF_r. As those sets of jobs are disjoint and $\frac{\alpha(\mathcal{J})}{T}$ is a lower bound on $\text{OPT}(\mathcal{J})$ (cf. Observation 2), these bounds lead to the term $\max\left\{\frac{4}{3r}, 2\right\}$ in the competitive ratio.

For sparse shelves, we distinguish whether there is a matching dense shelf of the same height (*non-singular* sparse shelves) or whether a sparse shelf is the only non-closed shelf of the respective height (making it a *singular* sparse shelf). While we can bound the energy demand contributed by non-singular sparse shelves in conjunction with their respective dense shelves in most cases, there does not need to exist a similar area bound on the height of singular sparse shelves. We achieve a bound nonetheless through an involved case distinction based on t, the point with maximum load, and r, the shelf-height parameter. On a high level, we bound their total load either in terms of the covered area if the point of justification is "far" from t, e.g., if $t \approx T$ and the shelves are left-justified, or in terms of $e_{\max}$, the maximum energy demand. In the latter case, we observe that only every other shelf height can have a sparse shelf contributing to the load, which allows us to bound their contribution to the total load by $\left(1 + \frac{r}{1-r^2}\right)e_{\max}$, accounting for the second term in the competitive ratio.

Before proving Theorem 1, we need some more notation: Recall that closed shelves are considered neither dense nor sparse; we use $\mathcal{C}$ to denote the set of jobs in closed shelves. From now on, unless stated explicitly, "shelf" always refers to a shelf that is *not* closed.

Let $\mathcal{D}$ and $\mathcal{S}$ denote the set of narrow jobs assigned to dense and sparse shelves, respectively. We additionally use $\mathcal{S}_{ns}$ for the subset of jobs assigned to non-singular sparse shelves, and $\mathcal{S}_s$ is the set of jobs assigned to singular sparse shelves. Let $\mathcal{D}^{\mathcal{L}}$ and $\mathcal{S}^{\mathcal{L}}$ be the subsets of jobs from $\mathcal{D}$ and $\mathcal{S}$, respectively, assigned to non-closed, left-justified shelves, and let $\mathcal{R}$ be the set of jobs in non-closed, right-justified shelves. Hence, we can partition $\mathcal{J}$, e.g., as $\mathcal{J} = \mathcal{W} \cup \mathcal{C} \cup \mathcal{R} \cup \mathcal{D}^{\mathcal{L}} \cup \mathcal{S}^{\mathcal{L}}$.

We use $H(\mathcal{J}')$ to denote the total effective height of all (also closed) *shelves* with jobs in $\mathcal{J}' \subseteq \mathcal{J}$. For ease of notation, let $H(\mathcal{W})$ denote the total energy demand of wide jobs. Let $\alpha(\mathcal{J}')$ denote the total area of jobs in $\mathcal{J}'$, i.e., $\alpha(\mathcal{J}') := \sum_{j \in \mathcal{J}'} p_j \cdot e_j$. Since any feasible solution must schedule all jobs, the following observation is trivial.

▶ **Observation 2.** *For any set $\mathcal{J}$ and any interval of width T, we have $\text{OPT}(\mathcal{J}) \geq \frac{\alpha(\mathcal{J})}{T}$.*

Wide Jobs and Closed Shelves. For wide jobs and jobs in closed shelves, we show that their total area is sufficiently large. Since any wide job has processing time greater than $\frac{T}{2}$, the next observation follows immediately.

▶ **Observation 3.** $H(\mathcal{W}) < \frac{2\alpha(\mathcal{W})}{T}$

For jobs in closed shelves, our closing rule ensures the same bound.

▶ **Proposition 4.** $H(\mathcal{C}) < \frac{2\alpha(\mathcal{C})}{T}$

Proof. We show the statement for each closed shelf individually: Fix a closed shelf and let j and j' be its two jobs. Without loss of generality, we can assume that $e_j \leq e_{j'}$. Since the shelf was closed, $p_j + p_{j'} > \left(1 - \frac{r}{2}\right)T$. As $p_j \leq \frac{T}{2}$, this implies $p_{j'} > \left(\frac{1}{2} - \frac{r}{2}\right)T$. Using further that $e_j > re_{j'}$, we can bound the area of j and j' by

$$\begin{aligned}\frac{e_j p_j + e_{j'} p_{j'}}{T} &> \left(1 - \frac{r}{2}\right) e_j + \left(\frac{1}{2} - \frac{r}{2}\right)(e_{j'} - e_j)\\ &> \frac{1}{2} r e_{j'} + \left(\frac{1}{2} - \frac{r}{2}\right) e_{j'}\\ &= \frac{e_{j'}}{2},\end{aligned}$$

as required. Summing over all jobs in closed shelves shows the statement. ◀

Dense and Sparse Shelves. In the following, we use the term *width* of a shelf to refer to the total processing time of jobs assigned to a given shelf. Next, we prove strong lower bounds on the widths of almost all dense and sparse shelves, showcasing the power of our closing rule. Recall that ACFF_r closes a shelf if it contains exactly two jobs and they have total processing time greater than $\left(1 - \frac{r}{2}\right)T$.

▶ **Lemma 5.** *Let $r \geq \frac{2}{3}$. Then, the following bounds hold:*
1. *Among all dense shelves, only the newest dense shelf of a given height can have width at most $\left(\frac{1}{2} + \frac{r}{4}\right)T$.*
2. *There are at most two dense shelves of a given height with width at most $\frac{3}{4}T$.*
3. *There is at most one shelf of a given height with width in $\left(\left(1 - \frac{r}{2}\right)T, \frac{3}{4}T\right]$.*

Since the shelves with width in $\left(\left(1 - \frac{r}{2}\right)T, \frac{3}{4}T\right]$ require special treatment when bounding the maximum load, we call such shelves *problematic*.

Proof. Fix a shelf height. We prove the statements in order.
1. Consider the oldest dense shelf that has width at most $\left(\frac{1}{2} + \frac{r}{4}\right)T$. It could have accommodated further jobs of processing time up to $\left(\frac{1}{2} - \frac{r}{4}\right)T$. Hence, any newer shelf only contains jobs of processing time greater than $\left(\frac{1}{2} - \frac{r}{4}\right)T$. Further, any newer *dense* shelf contains at least two such jobs. However, any shelf receiving the second job of processing time greater than $\left(\frac{1}{2} - \frac{r}{4}\right)T$ would be closed. Therefore, the oldest dense shelf of width at most $\left(\frac{1}{2} + \frac{r}{4}\right)T$ is in fact the newest dense shelf.
2. Let z be the number of dense shelves of width at most $\frac{3}{4}T$. If $z \leq 1$, the claim follows. Suppose $z \geq 2$ and consider the second newest dense shelf of width at most $\frac{3}{4}T$. Using the previous point and $r \geq \frac{2}{3}$, its width is greater than

$$\left(\frac{1}{2} + \frac{r}{4}\right)T \geq \left(1 - \frac{r}{2}\right)T.$$

Since the shelf is not closed, it must contain at least three jobs and, thus, at least one job of processing time at most $\frac{T}{4}$. Consequently, all older dense shelves have width greater than $\frac{3}{4}T$. Hence, $z = 2$.

3. Consider the newest problematic shelf. Since it is not closed, it has at least three jobs. Thus, it contains at least one job with processing time at most $\frac{T}{4}$. Hence, all older shelves have width greater than $\frac{3}{4}T$ and are not problematic. ◀

We now use the previous lemma to show that the jobs in dense and sparse shelves have enough area, even when adding a problematic shelf from a different shelf height.

▶ **Lemma 6.** *Let $r \in \left[\frac{2}{3}, 1\right)$ and let $k \in \mathbb{Z}$. If the problematic shelf of height r^k and the sparse shelf of height r^{k+1} are active at the same time, then the jobs in these two shelves and in the dense shelves of height r^{k+1} have an area greater than $\frac{3}{4}(H+h)rT$, where H is the total height of those dense shelves and $h \le r^k$ is the effective height of the problematic shelf.*

Proof. Fix $k \in \mathbb{Z}$. We will show a slightly stronger statement, namely, that the jobs in the problematic and the sparse shelf as well as the jobs in dense shelves of width at most $\frac{3}{4}T$ already guarantee an area greater than $\frac{3}{4}(H'+h)rT$, where H' is the height of those “thin” dense shelves. Clearly, since the jobs in any dense shelf of height r^{k+1} and width greater than $\frac{3}{4}T$ have total area greater than $\frac{3}{4}r^{k+2}T$, for such shelves, we immediately obtain a lower bound of $\frac{3}{4}(H-H')rT$ on their area.

It remains to consider the problematic and the sparse shelf as well as the dense shelves of height r^{k+1} and width at most $\frac{3}{4}T$. Let z denote the number of such shelves; by Lemma 5, we have $z \le 2$. We now distinguish three cases based on $z \in \{0, 1, 2\}$.

If $z = 0$, it suffices to argue that the jobs in the sparse shelf and the problematic shelf have an area greater than $\frac{3}{4}hrT$. Using that these two shelves together are wider than T and that the problematic shelf has width greater than $\left(1-\frac{r}{2}\right)T$ and is the taller of the two, we can lower bound the total area of their jobs by

$$\left(1-\frac{r}{2}\right)r^{k+1}T + \left(1-\left(1-\frac{r}{2}\right)\right)r^{k+2}T = \left(1-\frac{r}{2}+\frac{r^2}{2}\right)r^{k+1}T > \frac{3}{4}hrT.$$

If $z = 1$, we observe that the dense shelf and the sparse shelf together are wider than T. Hence, their jobs contribute at least $r^{k+2}T$ to the area. The jobs in the problematic shelf contribute at least $\left(1-\frac{r}{2}\right)r^{k+1}T$. We want to compare to a height of at most $r^k + r^{k+1}$ in this case, i.e.,

$$\frac{r^{k+2}T + \left(1-\frac{r}{2}\right)r^{k+1}T}{r^k + r^{k+1}} = \frac{r+1-\frac{r}{2}}{1+r}rT > \frac{3}{4}rT,$$

as required.

Finally, if $z = 2$, Lemma 5 guarantees that at least one of the two dense shelves has width greater than $\left(\frac{1}{2}+\frac{r}{4}\right)T$ and that its jobs contribute at least $\left(\frac{1}{2}+\frac{r}{4}\right)r^{k+2}T$ area. Combining the other dense shelf again with the sparse shelf to obtain a total width greater than T and an area of at least $r^{k+2}T$, we can bound the ratio again by

$$\frac{\left(\frac{1}{2}+\frac{r}{4}\right)r^{k+2}T + r^{k+2}T + \left(1-\frac{r}{2}\right)r^{k+1}T}{2r^{k+1}+r^k} = \frac{\left(\frac{1}{2}+\frac{r}{4}\right)r + r + 1 - \frac{r}{2}}{2r+1}rT > \frac{3}{4}rT,$$

which completes the proof. ◀

We remark that the proof above does not require that the jobs in the problematic shelf contribute an area greater than $\frac{3}{4}hrT$. Thus, analogous statements also hold when the problematic shelf has *lower* height than the other shelves and when it is not present at all.

▶ **Corollary 7.** *Let $r \in \left[\frac{2}{3}, 1\right)$ and let $k \in \mathbb{Z}$. If a problematic shelf of height r^k and a sparse shelf of height r^{k-1} are active at the same point in time, then the jobs in the problematic shelf, the sparse shelf, and the dense shelves of height r^{k-1} together have an area greater than $\frac{3}{4}(H+h)rT$, where H is the total height of those dense shelves and $h \le r^k$ is the effective height of the problematic shelf.*

▶ **Corollary 8.** *Let $r \in \left[\frac{2}{3}, 1\right)$ and consider a shelf height r^k for $k \in \mathbb{Z}$. The jobs in the dense and sparse shelves of that height have an area greater than $\frac{3}{4}HrT$, where H is the total height of the dense shelves.*

Sparse and Problematic Shelves. The remaining key technical insight for the proof of Theorem 1 is the observation that left- and right-justifying shelves allows us to give a better bound on the load contributed by sparse shelves for Online DSP than the bound that is possible for Online Strip Packing. Moreover, under some mild conditions, we are able to "hide" shelves of the opposite justification among them.

When applying the following lemma, the sparse left-justified shelves and problematic right-justified shelves will play the roles of $\mathcal{A}$ and $\mathcal{B}$ or vice versa, depending on the justification of the shelf with greatest height.

▶ **Lemma 9.** *Let $r \in (0,1)$ and consider $t \in [0,T]$. Further, let $\mathcal{A}$ and $\mathcal{B}$ be two sets of jobs in shelves of opposite justification such that $\mathcal{A}$ and $\mathcal{B}$ contain jobs of at most one shelf per shelf height. Further, suppose that a shelf with overall greatest height has the same justification as the shelves with jobs in $\mathcal{A}$. If, for each $k \in \mathbb{Z}$ where the shelves of height r^k have the same justification as the shelves in $\mathcal{A}$, at most one shelf of height r^k or of height r^{k+1} contributes to the load at t, then*

$$\mathrm{load}_{\mathcal{A}}(t) + \mathrm{load}_{\mathcal{B}}(t) \le \left(1 + \frac{r}{1-r^2}\right) \textsc{Opt}(\mathcal{J}).$$

Proof. Recall that $r < 1$ which implies $r^{k+1} < r^k$ for $k \in \mathbb{Z}$. Let $e_{\max} := \max_{j \in \mathcal{N}} e_j$ be the effective height of the greatest shelf and let k^* be such that $e_{\max} \in (r^{k^*+1}, r^{k^*}]$.

Now consider the shelves contributing to the total energy demand at t and consider $k \in \mathbb{Z}$ leading to the same justification as the shelves with jobs in $\mathcal{A}$. By assumption, we can pair the shelf of height r^k in $\mathcal{A}$ with the shelf of height r^{k+1} in $\mathcal{B}$ (if both exist) and at most one of these two shelves contributes to the load at t. If at most one exists, trivially at most one contributes. Hence, we can upper bound their total contribution either by r^k or by $e_{\max}$ if $k = k^*$. Thus,

$$\begin{aligned}\mathrm{load}_{\mathcal{A}}(t) + \mathrm{load}_{\mathcal{B}}(t) &\le e_{\max} + \sum_{i=1}^{\infty} r^{k^*+2i} = e_{\max} + r^{k^*}\frac{r^2}{1-r^2} = e_{\max} + r^{k^*+1}\frac{r}{1-r^2}\\ &\le \left(1 + \frac{r}{1-r^2}\right) e_{\max},\end{aligned}$$

where we used that $r^{k^*+1} < e_{\max}$ by assumption in the last step. To conclude the proof, it remains to observe that $e_{\max}$ is a lower bound on $\textsc{Opt}(\mathcal{J})$, that is,

$$\mathrm{load}_{\mathcal{A}}(t) + \mathrm{load}_{\mathcal{B}}(t) \le \left(1 + \frac{r}{1-r^2}\right) e_{\max} \le \left(1 + \frac{r}{1-r^2}\right) \textsc{Opt}(\mathcal{J}),$$

as required. ◀

Proof of Theorem 1. We are now ready to prove our main result: the upper bound on the competitive ratio of $\textsc{acFF}_r$. We restate Theorem 1 for convenience.

▶ **Theorem 1.** *There is a parameter $\varphi \in \left[\frac{2}{3}, 1\right]$ such that, for any constant $r \in [\varphi, 1)$ and any job list $\mathcal{J}$,* ALTERNATING AND CLOSING FIRST FIT$_r$ *achieves maximum energy demand*

$$\textsc{acFF}_r(\mathcal{J}) \leq \left(\max\left\{\frac{4}{3r}, 2\right\} + 1 + \frac{r}{1-r^2}\right) \textsc{Opt}(\mathcal{J}) = \left(3 + \frac{r}{1-r^2}\right) \textsc{Opt}(\mathcal{J}).$$

For $r = \varphi$, algorithm $\textsc{acFF}_r$ achieves a competitive ratio of approximately 4.263.

Proof. We consider the schedule produced by $\textsc{acFF}_r$ on an arbitrary instance $\mathcal{J}$. We will later give a concrete value of φ such that the theorem holds for all $r \in [\varphi, 1)$; for now, we only assume $\varphi \geq \frac{2}{3}$. Hence, all observations made so far are applicable.

We assume that the peak demand, i.e., the maximum load, occurs at $t \in \left[0, \frac{T}{2}\right]$. The case where peak demand occurs in $\left[\frac{T}{2}, T\right]$ can be handled analogously despite $\textsc{acFF}_r$ not behaving perfectly symmetrically: The only major break in symmetry is due to the justification of shelves depending on the parity of the exponent. Whenever we apply Lemma 9, we note that its upper bound is independent of the translation of parity into justification. In fact, the lemma can still be applied if $\mathcal{A} = \emptyset$ by interpreting “the same justification as $\mathcal{A}$” to mean “the opposite justification of $\mathcal{B}$”. In order to apply the arguments relating area to height, i.e., invoking one of Observation 3, Proposition 4, Lemmas 5 and 6, and Corollaries 7 and 8 also to $\left[\frac{T}{2}, T\right]$, one needs to exchange left-justified with right-justified and vice versa.

Therefore, it is sufficient to distinguish three cases, depending on whether t is in the interval $\left[0, \frac{T}{4}\right]$, $\left(\frac{T}{4}, \frac{rT}{2}\right)$, or $\left[\frac{rT}{2}, \frac{T}{2}\right]$.

Case 1: $t \in \left[0, \frac{T}{4}\right]$. In order to bound the load at t, we distinguish the load contributed by wide jobs (upper bounded by $H(\mathcal{W})$), jobs in closed shelves (upper bounded by $H(\mathcal{C})$), and the remaining jobs in sparse or dense, left- or right-justified shelves, that is,

$$\mathrm{load}(t) \leq H(\mathcal{W}) + H(\mathcal{C}) + \mathrm{load}_{\mathcal{R}}(t) + H(\mathcal{D}^{\mathcal{L}}) + \mathrm{load}_{\mathcal{S}^{\mathcal{L}}}(t).$$

By Observation 3 and Proposition 4, we can upper bound the first two terms by $2\frac{\alpha(\mathcal{W} \cup \mathcal{C})}{T}$. Further, any right-justified shelf active at t has width at least $\frac{3}{4}T$, and any job assigned to it has energy demand at least r times the height of the shelf, implying that the third term can be upper bounded by $\frac{4}{3r}\frac{\alpha(\mathcal{R})}{T}$. Using Corollary 8 for bounding $H(\mathcal{D}^{\mathcal{L}})$ we get

$$\mathrm{load}(t) \leq 2\frac{\alpha(\mathcal{W} \cup \mathcal{C})}{T} + \frac{4}{3r}\frac{\alpha(\mathcal{R})}{T} + \frac{4}{3r}\frac{\alpha(\mathcal{D}^{\mathcal{L}} \cup \mathcal{S}^{\mathcal{L}})}{T} + \mathrm{load}_{\mathcal{S}^{\mathcal{L}}}(t).$$

Depending on the justification of the shelf with greatest height, we set $\mathcal{A} = \mathcal{S}^{\mathcal{L}}$ and $\mathcal{B} = \emptyset$ or vice versa to apply Lemma 9 to the last term to obtain

$$\mathrm{load}(t) \leq 2\frac{\alpha(\mathcal{W} \cup \mathcal{C})}{T} + \frac{4}{3r}\frac{\alpha(\mathcal{R})}{T} + \frac{4}{3r}\frac{\alpha(\mathcal{D}^{\mathcal{L}} \cup \mathcal{S}^{\mathcal{L}})}{T} + \left(1 + \frac{r}{1-r^2}\right) \textsc{Opt}(\mathcal{J}).$$

Using that $\mathcal{W} \cup \mathcal{C}$, $\mathcal{R}$, and $\mathcal{D}^{\mathcal{L}} \cup \mathcal{S}^{\mathcal{L}}$ are disjoint and that $\frac{\alpha(\mathcal{J})}{T} \leq \textsc{Opt}(\mathcal{J})$, we conclude with

$$\mathrm{load}(t) \leq \left(\max\left\{\frac{4}{3r}, 2\right\} + 1 + \frac{r}{1-r^2}\right) \textsc{Opt}(\mathcal{J}).$$

Case 2: $t \in \left(\frac{T}{4}, \frac{rT}{2}\right)$. For every shelf height where the corresponding shelves are right-justified, there is at most one right-justified *problematic* shelf with width in $\left((1-\frac{r}{2})T, \frac{3}{4}T\right]$ by Lemma 5. We now distinguish two cases, based on the justification of the highest shelf, i.e., the shelf with largest height.

Case 2(L). Suppose that the highest shelf is left-justified. We partition the set $\mathcal{P}$ of right-justified problematic shelves that are active at t into $\mathcal{P}_1$ and $\mathcal{P}_2$ as follows: If the left-justified sparse shelf of height r^{k-1} is active at t, then the problematic shelf of height r^k belongs to $\mathcal{P}_1$; otherwise, it belongs to $\mathcal{P}_2$.
We again bound the peak demand by distinguishing the contributions of different shelf types. This time, we additionally handle $\mathcal{P}_1$, $\mathcal{P}_2$, and $\mathcal{R} \setminus \mathcal{P}$ separately:

$$\mathrm{load}(t) \leq H(\mathcal{W}) + H(\mathcal{C}) + H(\mathcal{D}^{\mathcal{L}}) + H(\mathcal{P}_1) + \mathrm{load}_{\mathcal{R}\setminus\mathcal{P}}(t) + \mathrm{load}_{\mathcal{P}_2}(t) + \mathrm{load}_{\mathcal{S}^{\mathcal{L}}}(t).$$

With Observation 3 and Proposition 4, for $H(\mathcal{W})$ and $H(\mathcal{C})$, and Corollary 7, for $H(\mathcal{D}^{\mathcal{L}})$ and $H(\mathcal{P}_1)$, we obtain that

$$\mathrm{load}(t) \leq 2\frac{\alpha(\mathcal{W} \cup \mathcal{C})}{T} + \frac{4}{3r}\frac{\alpha(\mathcal{D}^{\mathcal{L}} \cup \mathcal{S}^{\mathcal{L}} \cup \mathcal{P}_1)}{T} + \mathrm{load}_{\mathcal{R}\setminus\mathcal{P}}(t) + \mathrm{load}_{\mathcal{P}_2}(t) + \mathrm{load}_{\mathcal{S}^{\mathcal{L}}}(t).$$

Since any non-problematic right-justified shelf active at t has width greater than $\frac{3}{4}T$, we also get that $\mathrm{load}_{\mathcal{R}\setminus\mathcal{P}}(t) \leq \frac{4}{3r}\frac{\alpha(\mathcal{R}\setminus\mathcal{P})}{T}$. Hence,

$$\mathrm{load}(t) \leq 2\frac{\alpha(\mathcal{W} \cup \mathcal{C})}{T} + \frac{4}{3r}\frac{\alpha(\mathcal{D}^{\mathcal{L}} \cup \mathcal{S}^{\mathcal{L}} \cup \mathcal{P}_1 \cup (\mathcal{R}\setminus\mathcal{P}))}{T} + \mathrm{load}_{\mathcal{P}_2}(t) + \mathrm{load}_{\mathcal{S}^{\mathcal{L}}}(t).$$

Since the sets are again disjoint, we can upper bound the total area by $\alpha(\mathcal{J})$ and obtain

$$\mathrm{load}(t) \leq \max\left\{\frac{4}{3r}, 2\right\}\frac{\alpha(\mathcal{J})}{T} + \mathrm{load}_{\mathcal{P}_2}(t) + \mathrm{load}_{\mathcal{S}^{\mathcal{L}}}(t).$$

Using that $\frac{\alpha(\mathcal{J})}{T} \leq \textsc{Opt}(\mathcal{J})$ and applying Lemma 9 to $\mathrm{load}_{\mathcal{P}_2}(t) + \mathrm{load}_{\mathcal{S}^{\mathcal{L}}}(t)$ with $\mathcal{A} = \mathcal{S}^{\mathcal{L}}$ and $\mathcal{B} = \mathcal{P}_2$, we finally get

$$\mathrm{load}(t) \leq \left(\max\left\{\frac{4}{3r}, 2\right\} + 1 + \frac{r}{1-r^2}\right)\textsc{Opt}(\mathcal{J}).$$

Case 2(R). Suppose that the highest shelf is right-justified. We partition the set $\mathcal{P}$ (of right-justified problematic shelves active at t) again into two parts but, this time, according to the following rule: A right-justified problematic shelf of any height r^k that is active at t is in $\mathcal{P}_1$ if the sparse shelf of height r^{k+1} is active at t. Otherwise, the problematic shelf is in $\mathcal{P}_2$.
In analogy to Case 2(L), we bound the peak demand by

$$\mathrm{load}(t) \leq H(\mathcal{W}) + H(\mathcal{C}) + H(\mathcal{D}^{\mathcal{L}}) + H(\mathcal{P}_1) + \mathrm{load}_{\mathcal{R}\setminus\mathcal{P}}(t) + \mathrm{load}_{\mathcal{P}_2}(t) + \mathrm{load}_{\mathcal{S}^{\mathcal{L}}}(t).$$

As before, we again apply Observation 3 and Proposition 4, and now Lemma 6, to bound the first four terms by the area of the respective sets and use $\mathrm{load}_{\mathcal{R}\setminus\mathcal{P}}(t) \leq \frac{4}{3r}\frac{\alpha(\mathcal{R}\setminus\mathcal{P})}{T}$ to obtain

$$\begin{aligned}\mathrm{load}(t) &\leq \max\left\{\frac{4}{3r}, 2\right\}\frac{\alpha(\mathcal{W} \cup \mathcal{C} \cup \mathcal{D}^{\mathcal{L}} \cup \mathcal{S}^{\mathcal{L}} \cup \mathcal{P}_1 \cup (\mathcal{R}\setminus\mathcal{P}))}{T} + \mathrm{load}_{\mathcal{P}_2}(t) + \mathrm{load}_{\mathcal{S}^{\mathcal{L}}}(t) \\ &\leq \left(\max\left\{\frac{4}{3r}, 2\right\} + 1 + \frac{r}{1-r^2}\right)\textsc{Opt}(\mathcal{J}),\end{aligned}$$

where we used that the sets involved in the area bound are again disjoint, $\frac{\alpha(\mathcal{J})}{T} \leq \textsc{Opt}(\mathcal{J})$, and chose $\mathcal{A} = \mathcal{P}_2$ and $\mathcal{B} = \mathcal{S}^{\mathcal{L}}$ in Lemma 9 for the final inequality.

Case 3: $t \in \left[\frac{rT}{2}, \frac{T}{2}\right]$. In this case, every shelf active at t has a width of at least wT, where $w := \frac{r}{2}$. In particular, the jobs of *every* shelf active at t contribute an area of at least $wrT \cdot h$, where h is the height of the shelf. This bound includes closed shelves as well as wide jobs active at t. Let $\mathcal{J}'$ contain the jobs in closed or non-closed shelves active at t and all wide jobs active at t. If $\frac{1}{wr} = \frac{2}{r^2} \leq \max\left\{\frac{4}{3r}, 2\right\} + 1 + \frac{r}{1-r^2}$, the jobs in $\mathcal{J}$' contribute enough area to use a single bound of the form

$$\text{load}(t) \leq H(\mathcal{J}') \leq \frac{1}{wr}\frac{\alpha(\mathcal{J}')}{T} \leq \left(\max\left\{\frac{4}{3r}, 2\right\} + 1 + \frac{r}{1-r^2}\right)\textsc{Opt}(\mathcal{J}).$$

Note that, since $r \geq \varphi \geq \frac{2}{3}$ by assumption, the condition $\frac{2}{r^2} \leq \max\left\{\frac{4}{3r}, 2\right\} + 1 + \frac{r}{1-r^2}$ is equivalent to $3r^4 - r^3 - 5r^2 + 2 \leq 0$. Hence, for $r \geq \varphi'$, the unique root of $3x^4 - x^3 - 5x^2 + 2$ in $[0, 1]$, the claim holds.

It remains to determine $\varphi \geq \frac{2}{3}$ and to consider $\varphi \leq r < \varphi'$. In this case,

$$h_r := \max\left\{\frac{4}{3r}, 2\right\} + 1 + \frac{r}{1-r^2} - \left(1 + \frac{1}{1-r}\right) = 2 - \frac{1}{1-r^2} > 0.$$

If $H(\mathcal{W}) + H(\mathcal{C}) + H(\mathcal{D}) \leq h_r\textsc{Opt}(\mathcal{J})$, we have enough room to simultaneously bound the combined contribution of all sparse shelves: Specifically,

$$\text{load}_{\mathcal{S}}(t) \leq \max_{j\in\mathcal{N}} e_j + \sum_{k=1}^{\infty} r^{k^*+k} = \max_{j\in\mathcal{N}} e_j + \frac{r^{k^*+1}}{1-r} \leq \left(1 + \frac{1}{1-r}\right)\textsc{Opt}(\mathcal{J}),$$

where k^* is the exponent of the largest shelf height r^{k^*}, and hence $r^{k^*+1} < \max_{j\in\mathcal{N}} e_j \leq \textsc{Opt}(\mathcal{J})$.

Now, suppose that $H(\mathcal{W}) + H(\mathcal{C}) + H(\mathcal{D}) > h_r\textsc{Opt}(\mathcal{J})$. We use $\hat{H}$ to denote the total height of singular sparse shelves active at t. Each such shelf has width at least wT, so overall the jobs in such shelves have area at least $rw\hat{H}T$. Since a non-singular sparse shelf combined with any dense shelf of the same height has width greater than T (and every dense shelf has width greater than $\frac{T}{2}$), we know that the jobs in these shelves have an area of at least $\frac{r(H(\mathcal{D})+H(\mathcal{S}_{ns}))T}{2}$. We additionally use Observation 3 and Proposition 4 to obtain

$$\textsc{Opt}(\mathcal{J}) \geq \frac{H(\mathcal{W})}{2} + \frac{H(\mathcal{C})}{2} + \frac{r(H(\mathcal{D}) + H(\mathcal{S}_{ns}))}{2} + wr\hat{H}.$$

We rearrange to obtain that

$$H(\mathcal{S}_{ns}) + \hat{H} \leq \frac{\textsc{Opt}(\mathcal{J})}{wr} - \frac{1}{2w}(H(\mathcal{W}) + H(\mathcal{C}) + H(\mathcal{D})).$$

Therefore, the peak demand is bounded by

$$\begin{aligned}\text{load}(t) &\leq H(\mathcal{W}) + H(\mathcal{C}) + H(\mathcal{D}) + H(\mathcal{S}_{ns}) + \hat{H}\\ &\leq \frac{\textsc{Opt}(\mathcal{J})}{wr} + \left(1 - \frac{1}{2w}\right)(H(\mathcal{W}) + H(\mathcal{C}) + H(\mathcal{D}))\\ &\leq \left(\frac{1}{wr} + \left(1 - \frac{1}{2w}\right)h_r\right)\textsc{Opt}(\mathcal{J}).\end{aligned}$$

Choosing φ as the unique root of $x^4 + x^3 - 4x^2 - x + 2$ in $[0, 1]$, we can bound this by

$$\leq \left(\max\left\{\frac{4}{3r}, 2\right\} + 1 + \frac{r}{1-r^2}\right)\textsc{Opt}(\mathcal{J}).$$

We note that $\varphi \approx 0.679$. Inserting $r = \varphi$ into the bound concludes the proof. ◀

▶ Remark 10. We observe that, in Case 3, several of the inequalities are not tight. With additional (technical) arguments, we can extend the analysis to $\bar{\varphi} \approx 0.674$, improving the competitive ratio slightly to approximately 4.237. Since this proof does not offer new insights, we do not give the details here.

We note that the use of left- and right-justified shelves does not improve the competitive ratio for Strip Packing, as upon arrival of a new job its starting time as well as its *starting height* need to be fixed. In particular, the creation of a shelf (independent of its justification) does increase the possible starting height for all later-created shelves by its height. In other words, in Strip Packing we *cannot* distinguish between the height and the effective height.

2.3 Lower Bound on the Competitive Ratio of $\textsc{acFF}_r$

Note that, even though we use left- and right-justified shelves and the closing rule, we can adapt the lower bound by Ye, Han, and Zhang [31] on their algorithm Revised-Shelf and give a lower bound on the competitive ratio of $\textsc{acFF}_r$. We give the proof in the appendix for the sake of completeness.

▶ **Lemma 11.** *Let $r \in (0,1)$. The competitive ratio of $\textsc{acFF}_r$ is at least $\frac{10r+5}{8r+2} + \frac{1}{1-r^2} > 2.910$.*

3 Lower Bound

As a warm-up, we start by proving a lower bound of $1 + \frac{\sqrt{2}}{2} \approx 1.707$ on the competitive ratio, which is already sufficient to separate Online DSP from Online MPAS, before explaining how to modify the instance to obtain a lower bound of approximately 1.812.

▶ **Theorem 12.** *The competitive ratio of any deterministic online algorithm* Alg *for Online Demand Strip Packing is at least $1 + \frac{\sqrt{2}}{2} \approx 1.707$.*

Proof. Let $T := 12$ and define $x := 2 + 2\sqrt{2}$. We first define the sequence of jobs considered. Let $\mathcal{J}_1$ be comprised of four jobs with processing time 3 and unit energy demand. Let $\mathcal{J}_2$ be comprised of two jobs with processing time 5 and energy demand $\frac{x}{2}$. Lastly, we let $\mathcal{J}_3$ consist of a single job with processing time 4 and energy demand x. We (potentially) concatenate these in increasing order of their index to obtain our instance; that is, we use, e.g., $\mathcal{J}_1\mathcal{J}_2$ to refer to an instance in which $\mathcal{J}_1$ and $\mathcal{J}_2$ are presented one after the other, in the order of their indices, to the online algorithm. It is easy to check that $\textsc{Opt}(\mathcal{J}_1) \leq 1$ by placing all jobs next to each other, $\textsc{Opt}(\mathcal{J}_1\mathcal{J}_2) \leq \frac{x}{2} + 1$ by scheduling the jobs in $\mathcal{J}_1\mathcal{J}_2$ as shown on the right in Figure 1, and $\textsc{Opt}(\mathcal{J}_1\mathcal{J}_2\mathcal{J}_3) \leq x$ as shown on the left in Figure 1.

We now consider the solution constructed by the online algorithm. First consider the jobs in $\mathcal{J}_1$. If the online algorithm schedules them such that at least two overlap (i.e., are active at the same time), then the resulting maximum load is at least 2. Since the optimal peak for these jobs is 1, the algorithm can thus be strictly 2-competitive at best:

$$\sup_{\mathcal{J}} \frac{\textsc{Alg}(\mathcal{J})}{\textsc{Opt}(\mathcal{J})} \geq \frac{\textsc{Alg}(\mathcal{J}_1)}{\textsc{Opt}(\mathcal{J}_1)} \geq 2.$$

Now assume that Alg schedules the jobs in $\mathcal{J}_1$ such that none of them overlap. By construction, this implies that the load at every point in the interval $[0, T]$ is exactly 1. We now release the jobs in $\mathcal{J}_2$. Again, the algorithm will either schedule the jobs in $\mathcal{J}_2$ such that they overlap or such that they do not. If the two jobs in $\mathcal{J}_2$ overlap, the maximum load of the resulting schedule is equal to $x + 1$, yielding a lower bound of

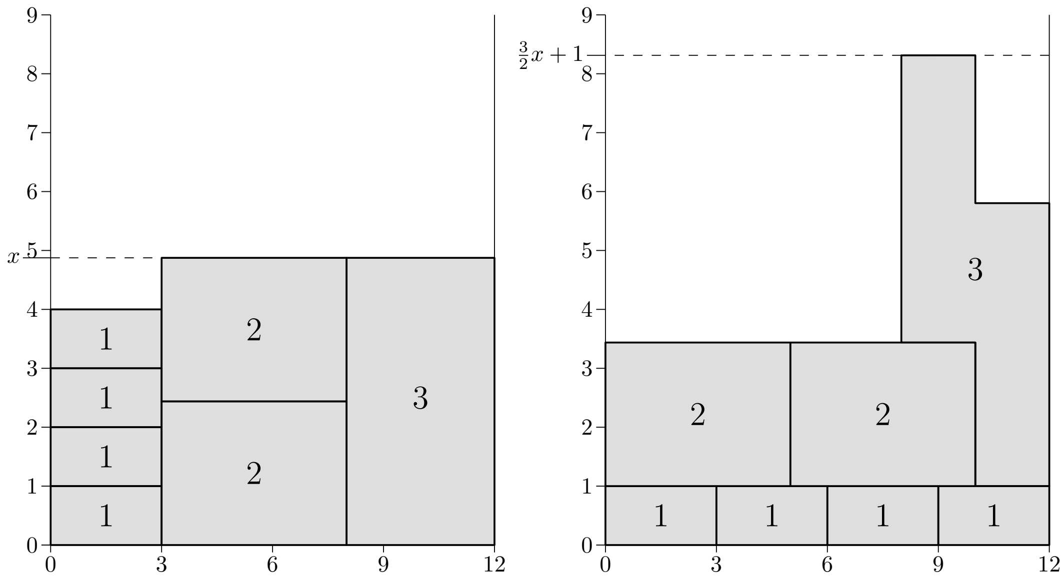


Figure 1 The lower bound of 1.707 for Online DSP. The indices of the jobs refer to the sets they are a part of. The optimal schedule (left) and the final schedule of any algorithm that attempts to be strictly 1.707-competitive (right).

$$\frac{\textsc{Alg}(\mathcal{J}_1\mathcal{J}_2)}{\textsc{Opt}(\mathcal{J}_1\mathcal{J}_2)} \geq \frac{x+1}{\frac{x}{2}+1} = 1 + \frac{\sqrt{2}}{2} \approx 1.707.$$

Now assume that the jobs in $\mathcal{J}_2$ are placed such that they do not overlap. By construction, they together occupy a width of 10 on the strip. We then release the final job, which must overlap with at least one of the jobs in $\mathcal{J}_2$. As a result, the maximum load of the algorithm's schedule is exactly $x + \frac{x}{2} + 1$ (see Figure 1), thus yielding a lower bound of

$$\frac{\textsc{Alg}(\mathcal{J}_1\mathcal{J}_2\mathcal{J}_3)}{\textsc{Opt}(\mathcal{J}_1\mathcal{J}_2\mathcal{J}_3)} \geq \frac{x+\frac{x}{2}+1}{x} = 1 + \frac{\sqrt{2}}{2} \approx 1.707.$$

In all cases, we have shown a lower bound of 1.707. ◀

Using a more carefully defined family of instances, consisting of up to five job types, we can prove that, in fact, no deterministic online algorithm can achieve a competitive ratio of 1.812. The family of instances can be seen in Figure 3. We defer the proof to Appendix B.

▶ **Theorem 13.** *The competitive ratio of any deterministic online algorithm* Alg *for Online Demand Strip Packing is at least ϱ for some $\varrho \in (1.812, 1.813)$.*

4 Conclusion

Even though Demand Strip Packing has been well-studied in the offline setting and has important applications in smart-grid scheduling, we are the first to investigate the problem in an online setting, in which scheduling decisions must be made before knowing the whole instance. We give an online algorithm, Alternating and Closing First Fit$_r$, reminiscent of shelf algorithms previously only employed for Online Strip Packing, and upper bound its competitive ratio by 4.263, which is significantly better than the best currently known guarantee for Online Strip Packing. We further give a lower bound of 1.812 on the competitive ratio of *any* deterministic online algorithm for Online DSP and thus separate the problem from a previously studied special case, Minimum Peak Appointment Scheduling, for which a competitive ratio of $\frac{5}{3}$ is possible.

References

1 Susanne Albers. Better bounds for online scheduling. *SIAM J. Comput.*, 29(2):459–473, 1999. doi:10.1137/S0097539797324874.

2 János Balogh, József Békési, György Dósa, Leah Epstein, and Asaf Levin. A new and improved algorithm for online bin packing. In *Proceedings of ESA 2018*, volume 112 of *LIPIcs*, pages 5:1–5:14, 2018. doi:10.4230/LIPIcs.ESA.2018.5.

3 János Balogh, József Békési, György Dósa, Jiří Sgall, and Rob van Stee. The optimal absolute ratio for online bin packing. *J. Comput. Syst. Sci.*, 102:1–17, 2019. doi:10.1016/j.jcss.2018.11.005.

4 János Balogh, György Dósa, Leah Epstein, and Łukasz Jeż. Lower bounds on the performance of online algorithms for relaxed packing problems. In *Proceedings of IWOCA 2022*, volume 13270 of *LNCS*, pages 101–113, 2022. doi:10.1007/978-3-031-06678-8_8.

5 János Balogh, József Békési, György Dósa, Leah Epstein, and Asaf Levin. A new lower bound for classic online bin packing. *Algorithmica*, 83(7):2047–2062, 2021. doi:10.1007/s00453-021-00818-7.

6 Yair Bartal, Amos Fiat, Howard J. Karloff, and Rakesh V. Vohra. New algorithms for an ancient scheduling problem. *J. Comput. Syst. Sci.*, 51(3):359–366, 1995. doi:10.1006/jcss.1995.1074.

7 Iwo Błądek, Maciej Drozdowski, Frédéric Guinand, and Xavier Schepler. On contiguous and non-contiguous parallel task scheduling. *J. Sched.*, 18(5):487–495, 2015. doi:10.1007/s10951-015-0427-z.

8 Max A. Deppert, Klaus Jansen, Arindam Khan, Malin Rau, and Malte Tutas. Peak demand minimization via sliced strip packing. *Algorithmica*, 85(12):3649–3679, 2023. doi:10.1007/s00453-023-01152-w.

9 Franziska Eberle, Felix Hommelsheim, Malin Rau, and Stefan Walzer. A tight $(3/2+\varepsilon)$-approximation algorithm for demand strip packing. In *Proceedings of SODA 2025*, pages 641–699, 2025. doi:10.1137/1.9781611978322.20.

10 Célia Escribe, Michael Hu, and Retsef Levi. Competitive algorithms for the online minimum peak job scheduling. *Oper. Res.*, 73(1):408–423, 2025. doi:10.1287/opre.2021.0080.

11 Rudolf Fleischer and Michaela Wahl. Online scheduling revisited. In *Proceedings of ESA 2000*, volume 1879 of *LNCS*, pages 202–210, 2000. doi:10.1007/3-540-45253-2_19.

12 Gábor Galambos and Gerhard J. Woeginger. An on-line scheduling heuristic with better worst case ratio than Graham's list scheduling. *SIAM J. Comput.*, 22(2):349–355, 1993. doi:10.1137/0222026.

13 Waldo Gálvez, Fabrizio Grandoni, Afrouz Jabal Ameli, and Kamyar Khodamoradi. Approximation algorithms for demand strip packing. In *Proceedings of APPROX/RANDOM 2021*, volume 207 of *LIPIcs*, pages 20:1–20:24, 2021. doi:10.4230/LIPIcs.APPROX/RANDOM.2021.20.

14 Ronald L. Graham. Bounds for certain multiprocessing anomalies. *Bell Syst. Tech. J.*, 45(9):1563–1581, 1966. doi:10.1002/j.1538-7305.1966.tb01709.x.

15 Xin Han, Kazuo Iwama, Deshi Ye, and Guochuan Zhang. Strip packing vs. bin packing. In *Proceedings of AAIM 2007*, volume 4508 of *LNCS*, pages 358–367, 2007. doi:10.1007/978-3-540-72870-2_34.

16 Johann L. Hurink and Jacob J. Paulus. Online algorithm for parallel job scheduling and strip packing. In *Proceedings of WAOA 2007*, volume 4927 of *LNCS*, pages 67–74, 2007. doi:10.1007/978-3-540-77918-6_6.

17 Klaus Jansen, Malin Rau, and Malte Tutas. Hardness and tight approximations of demand strip packing. In *Proceedings of SPAA 2024*, pages 479–489, 2024. doi:10.1145/3626183.3659971.

18 David S. Johnson. *Near-Optimal Bin Packing Algorithms*. PhD thesis, Massachusetts Institute of Technology, 1973. URL: https://hdl.handle.net/1721.1/57819.

19 David S. Johnson, Alan J. Demers, Jeffrey D. Ullman, Michael R. Garey, and Ronald L. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM J. Comput.*, 3(4):299–325, 1974. doi:10.1137/0203025.

20 David R. Karger, Steven J. Phillips, and Eric Torng. A better algorithm for an ancient scheduling problem. *J. Algorithms*, 20(2):400–430, 1996. doi:10.1006/jagm.1996.0019.

21 Chi Chung Lee and Der-Tsai Lee. A simple on-line bin-packing algorithm. *J. ACM*, 32(3):562–572, 1985. doi:10.1145/3828.3833.

22 Frank M. Liang. A lower bound for on-line bin packing. *Inf. Process. Lett.*, 10(2):76–79, 1980. doi:10.1016/S0020-0190(80)90077-0.

23 Gilbert M. Masters. *Renewable and Efficient Electric Power Systems*. John Wiley & Sons, 2013.

24 Georgios Moneftsis. Online demand strip packing. Master's thesis, Technische Universität Berlin, 2025.

25 John F. Rudin, III. *Improved bounds for the online scheduling problem*. PhD thesis, The University of Texas at Dallas, 2001. URL: https://www.proquest.com/openview/2bfed0b3f477b3dc40ef775ce4d0e6f4.

26 Steven S. Seiden. On the online bin packing problem. *J. ACM*, 49(5):640–671, 2002. doi:10.1145/585265.585269.

27 Devin Smedira and David Shmoys. Scheduling appointments online: The power of deferred decision-making. In *Proceedings of WAOA 2022*, volume 13538 of *LNCS*, pages 82–115, 2022. doi:10.1007/978-3-031-18367-6_5.

28 André van Vliet. An improved lower bound for on-line bin packing algorithms. *Inf. Process. Lett.*, 43(5):277–284, 1992. doi:10.1016/0020-0190(92)90223-I.

29 Andrew Chi-Chih Yao. New algorithms for bin packing. *J. ACM*, 27(2):207–227, 1980. doi:10.1145/322186.322187.

30 Sean Yaw, Brendan Mumey, Erin McDonald, and Jennifer Lemke. Peak demand scheduling in the smart grid. In *Proceedings of SmartGridComm 2014*, pages 770–775, 2014. doi:10.1109/SmartGridComm.2014.7007741.

31 Deshi Ye, Xin Han, and Guochuan Zhang. A note on online strip packing. *J. Comb. Optim.*, 17(4):417–423, 2009. doi:10.1007/s10878-007-9125-x.

32 Guosong Yu, Yanling Mao, and Jiaoliao Xiao. A new lower bound for online strip packing. *Eur. J. Oper. Res.*, 250(3):754–759, 2016. doi:10.1016/j.ejor.2015.10.012.

33 Guosong Yu, Yanling Mao, and Jiaoliao Xiao. New upper bounds for online strip packing. *Discret. Optim.*, 23:20–32, 2017. doi:10.1016/j.disopt.2016.11.004.

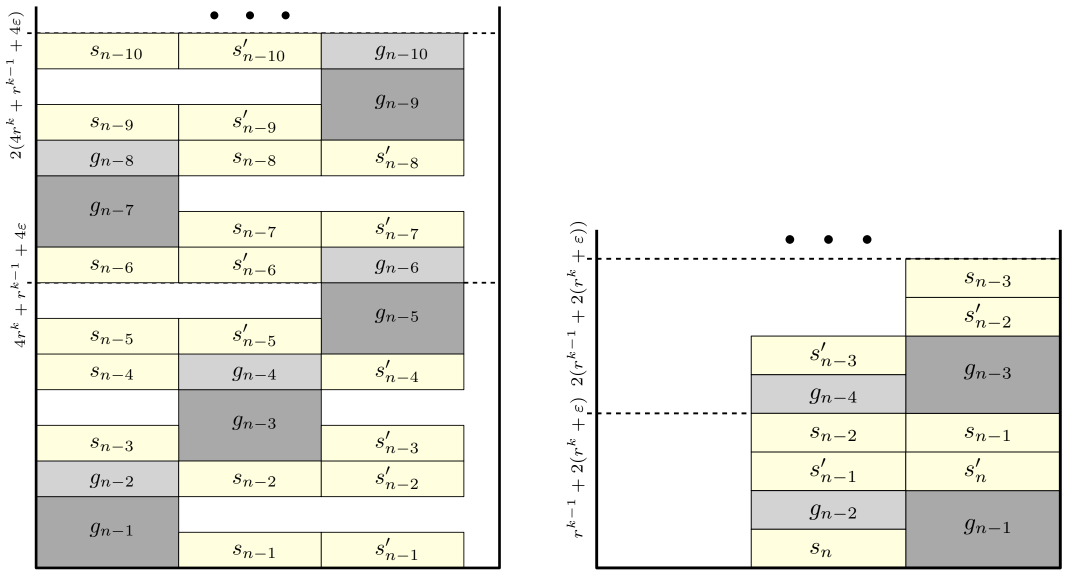
A Omitted Proofs from Section 2.3

Lower Bound for $\textsc{acFF}_r$

In the following, we adapt the lower bound by Ye, Han, and Zhang [31] on their algorithm Revised-Shelf in order to give a lower bound on the competitive ratio of $\textsc{acFF}_r$.

▶ **Lemma 11.** *Let $r \in (0,1)$. The competitive ratio of $\textsc{acFF}_r$ is at least $\frac{10r+5}{8r+2} + \frac{1}{1-r^2} > 2.910$.*

Proof. Let $k \in 2\mathbb{Z}+1$ be a sufficiently large odd integer. Let $\varepsilon \in (0, r^k - r^{k+1})$ (which will eventually be considered in the limit $\varepsilon \to 0$), let $\delta \in (0, \frac{6\varepsilon}{k})$, and let $\Delta > \varepsilon$. Define $H \coloneqq r + \varepsilon$, and let $n \in (5\mathbb{Z}+1) \cap (2\mathbb{Z}+1)$, i.e., $n-1$ is an integer multiple of 5 and n is odd, be maximal such that $\frac{n-1}{5} \cdot (4r^k + r^{k-1} + 4\varepsilon) + r^k + \varepsilon \leq H$.



(a) A possible schedule of growing jobs g_i, and shrinking jobs s_i and s'_i. This pattern repeats after every ten triplets. Five consecutive triplets contribute demand at most $4r^k + r^{k-1} + 4\varepsilon$. Further, the gap on the right is wide enough for the thin jobs. We use $r \approx \frac{1}{2}$ in this simplified figure.

(b) Schedule produced by ACFF$_r$ on the first twelve jobs of the sequence of growing/shrinking jobs. The pattern repeats after every six jobs, using the same jobs (and again $r \approx \frac{1}{2}$) as on the left. Note that all tall growing jobs finish at the right end of the interval.

Figure 2 The optimal schedule and the schedule produced by ACFF$_r$ side by side.

We construct an instance with $\frac{k-1}{2} + 3n - 1$ jobs. Each job can be thought of as having one of three "types": *thin*, *shrinking*, or *growing*. First, $\frac{k-1}{2}$ *thin* jobs arrive in order of increasing height. The processing time of a thin job is δ and its energy demand is in $\{r^{k-2} + \varepsilon, r^{k-4} + \varepsilon, \ldots, r^3 + \varepsilon, r + \varepsilon\}$.

Then, $3n - 1$ shrinking/growing jobs arrive, specified as follows. A *shrinking* job, denoted s_j and s'_j, for $j \in \{n, n-1, \ldots, 1\}$, has processing time $p^s_j := \frac{1}{3} + 2^{2j}\Delta - \varepsilon$ and energy demand $e^s_j := r^k + \varepsilon$. Note that the width of shrinking jobs decreases as j decreases. A *growing* job, denoted g_j, for $j \in \{n-1, n-2, \ldots, 1\}$, has processing time $p^g_j := \frac{1}{3} - 2^{2j+1}\Delta - \varepsilon$ and energy demand $e^g_j := r^{k-1}$ if j is even (g_j is *tall*) and $e^g_j := r^k + \varepsilon$ if j is odd (g_j is *short*). The width of growing jobs *increases* as j decreases.

Now, we make two observations: First, the jobs g_j, s_j, and s'_j (a *triplet*) fit next to each other, occupying a width of $1 - 3\varepsilon$. Second, the thin jobs in $\mathcal{T}$ can also be scheduled next to each other, occupying a total width of $\frac{k-1}{2}\delta < 3\varepsilon$ and height at most H.

By arranging the tall growing jobs in a diagonal structure (see Figure 2a), all of the triplets can be placed such that five rows have total load at most $4r^k + r^{k-1} + 4\varepsilon$. These triplet rows alternate with respect to whether they contain a tall or a short growing jobs. There are two remaining shrinking jobs, s_n and s'_n, that do not occur in a triplet. They are placed in a row on top of the triplets. Overall, this results in a peak demand of at most

$$\textsc{Opt}(\mathcal{J}) \leq \max\left\{H, \frac{n-1}{5} \cdot \left(4r^k + r^{k-1} + 4\varepsilon\right) + r^k + \varepsilon\right\} = H.$$

However, if the jobs arrive in adversarial order, ACFF$_r$ incurs a higher peak: We first release all thin jobs before the growing/shrinking jobs. Due to the height choices for the thin jobs, they will finish at time 1 as they belong to every other shelf, resulting in

$$\sum_{i=0}^{\frac{k-1}{2}-1} (r^{2i+1}+\varepsilon) = \frac{k-1}{2}\cdot\varepsilon + \frac{r-r^k}{1-r^2} \xrightarrow{k\to\infty,\varepsilon\to 0} \frac{r}{1-r^2} \tag{1}$$

demand in the interval $[1-\delta, 1)$.

Consider now the shrinking and growing jobs: They all belong to the same shelf height r^{k-1} as $e_j^s, e_j^g \in (r^k, r^{k-1}]$. Hence, they will all be assigned to right-justified shelves. In particular, each such shelf will overlap with the thin jobs in the interval $[1-\delta, 1)$. When releasing the jobs in the following order

$$g_{n-1}, s_n, s'_n, g_{n-2}, s_{n-1}, s'_{n-1}, g_{n-3}, \dots, g_1, s_2, s'_2, s_1, s'_1,$$

$\textsc{acFF}_r$ will only assign two consecutive jobs to the same shelf: Since $p_j^s > \frac{1}{3}$ for any $j \in [n]$, no three shrinking jobs can be assigned to the same shelf, and the only way to pair a growing job $g_{j'}$ with two shrinking jobs s_j and s'_j is if $j' \geq j$, which cannot happen in the above order. It remains to show that the algorithm cannot assign a growing job to a shelf already occupied by a growing and a shrinking job due to the order of arrival. Pairing a growing job g_j with a shrinking job s_{j+1} leaves at most

$$1-\left(p_{j+1}^s + p_j^g\right) = 1-\left(\frac{1}{3}+2^{2(j+1)}\Delta-\varepsilon+\frac{1}{3}-2^{2j+1}\Delta-\varepsilon\right) = \frac{1}{3}-2^{2j+1}\Delta+2\varepsilon$$

processing time in the same shelf. By definition, any shrinking job has processing time greater than $\frac{1}{3}$, implying that no shrinking job can be assigned to such a shelf. Similarly, any growing job $g_{j'}$ for $j' < j$, has processing time at least

$$\frac{1}{3}-2^{2j'+1}\Delta-\varepsilon \geq \frac{1}{3}-2^{2(j-1)+1}\Delta-\varepsilon = \frac{1}{3}-2^{2j-1}\Delta-\varepsilon > \frac{1}{3}-2^{2j+1}\Delta+2\varepsilon,$$

as $j \geq 1$ and $\Delta > \varepsilon$. Further, the growing jobs are scheduled alternatingly as late (finishing at 1) or as early (finishing at the starting time of the previous shrinking job) in their respective shelves (see Figure 2b). Starting with g_{n-1} as a late job, all tall growing jobs will finish at 1. Hence, six consecutive jobs contribute $r^{k-1} + 2\cdot(r^k+\varepsilon)$ to the peak demand. Thus, in total they contribute

$$\frac{n-1}{2}\cdot\left(r^{k-1}+2\cdot(r^k+\varepsilon)\right)+(r^k+\varepsilon),$$

where the last two shrinking jobs s_1 and s'_1 contribute the additive $(r^k+\varepsilon)$.

We can then lower bound the maximum demand of the schedule generated by the algorithm as follows: We chose n odd maximal such that $n-1$ is an (even) integer multiple of 5 with

$$\frac{n-1}{5}\cdot\left(4r^k+r^{k-1}+4\varepsilon\right)+r^k+\varepsilon \leq H.$$

Hence, $n-1$ is an integer multiple of 10 and the next larger such number is too big, i.e.,

$$\frac{(n-1)+10}{5}\cdot\left(4r^k+r^{k-1}+4\varepsilon\right)+r^k+\varepsilon > H = r+\varepsilon$$

$$\frac{(n-1)+10}{5}\cdot\left(4r^k+r^{k-1}+4\varepsilon\right) > r-r^k$$

$$(n-1)+10 > 5\cdot\frac{r-r^k}{4r^k+r^{k-1}+4\varepsilon}$$

$$n > \ell := -9+5\cdot\frac{r-r^k}{4r^k+r^{k-1}+4\varepsilon}.$$

Compared to the offline optimum H, we get the following ratio for the shrinking and growing jobs:

$$\lim_{k\to\infty}\lim_{\varepsilon\to 0}\frac{\frac{\ell-1}{2}\left(r^{k-1}+2(r^k+\varepsilon)\right)+r^k+\varepsilon}{H}=\lim_{k\to\infty}\lim_{\varepsilon\to 0}\frac{\frac{\ell-1}{2}\left(r^{k-1}+2(r^k+\varepsilon)\right)+r^k+\varepsilon}{r+\varepsilon}.$$

Now, using the definition of ℓ, we obtain after simple calculations

$$=\lim_{k\to\infty}\frac{10r^3+5r^2-10r^k-63r^{k+1}-82r^{k+2}}{8r^3+2r^2}$$

$$=\frac{10r^3+5r^2}{8r^3+2r^2}=\frac{10r+5}{8r+2}.$$

Recall from (1) that, in the limit, the thin jobs contribute $\frac{r}{1-r^2}$ to the maximum demand of the algorithm's schedule. Comparing the demand of the thin jobs to the offline optimum $H=r+\varepsilon$ then gives a ratio of $\frac{1}{1-r^2}$ in the limit. Together with the shrinking and growing jobs, this results in the promised lower bound of $\frac{10r+5}{8r+2}+\frac{1}{1-r^2}$. ◀

B Omitted Proofs from Section 3

In this section, we prove Theorem 13, which we restate for convenience.

▶ **Theorem 13.** *The competitive ratio of any deterministic online algorithm* Alg *for Online Demand Strip Packing is at least ϱ for some $\varrho \in (1.812, 1.813)$.*

Proof. We choose $T := 108$ and note that 108 is divisible by 27 and by 12. We fix an algorithm Alg for Online Demand Strip Packing and define an instance $\mathcal{J}$, which consists of up to five different job lists $\mathcal{J}_1,\dots,\mathcal{J}_5$, depending on the behavior of Alg on the sublists. Similar to the instance used in the proof of Theorem 12, we concatenate these job lists in the order of their indices if necessary. In contrast to the previous instance, we do not give the energy demands exactly, but rather provide a mathematical program to optimize over the energy demands.

The list $\mathcal{J}_1$ consists of 9 jobs with processing time $p_1 := 12$ and energy demand x_1. The list $\mathcal{J}_2$ consists of 6 jobs with processing time $p_2 := 18$ and energy demand x_2. The list $\mathcal{J}_3$ consists of 3 jobs with processing time $p_3 := 32$ and energy demand x_3. The list $\mathcal{J}_4$ contains 2 jobs with processing time $p_4 := 45$ and energy demand x_4. The final list $\mathcal{J}_5$ contains a single job with processing time $p_5 := 27$ and energy demand x_5. Let $\mathcal{J}$ denote the concatenation of all five lists, i.e., $\mathcal{J} := \mathcal{J}_1\cdots\mathcal{J}_5$.

In an optimal solution for $\mathcal{J}_1$ all jobs can be scheduled next to each other ensuring $\textsc{Opt}(\mathcal{J}_1)=x_1$. Again, all of jobs in $\mathcal{J}_2$ fit next to each other, implying that $\textsc{Opt}(\mathcal{J}_1\mathcal{J}_2)=x_1+x_2$. If $x_3\geq 9x_1$, then $\textsc{Opt}(\mathcal{J}_1\cdots\mathcal{J}_3)=x_2+x_3$. If $x_4\geq 6x_2$, then $\textsc{Opt}(\mathcal{J}_1\cdots\mathcal{J}_4)=x_3+x_4$. As $36=3p_1=2p_2>p_3$, we can schedule 3 jobs of $\mathcal{J}_1$ or 2 jobs of $\mathcal{J}_2$ or one job of $\mathcal{J}_3$ next to one job of $\mathcal{J}_4$ and the final job within $36+45+27=108=T$ units of time. Hence, if $3(x_1+x_2+x_3)\leq x_5$ and $2x_4\leq x_5$, then $\textsc{Opt}(\mathcal{J})=x_5$. In order to ensure that the respective optimal solutions are as claimed and as shown in Figure 3, we have just obtained several constraints on the energy demands, which we collect next.

$$\begin{aligned}9x_1&\leq x_3\\6x_2&\leq x_4\\2x_4&\leq x_5\\3(x_1+x_2+x_3)&\leq x_5,\end{aligned}\tag{C}$$

where the first two constraints ensure $\textsc{Opt}(\mathcal{J}_1\cdots\mathcal{J}_3)=x_2+x_3$ and $\textsc{Opt}(\mathcal{J}_1\cdots\mathcal{J}_4)=x_3+x_4$, respectively, and the last two constraints ensure $\textsc{Opt}(\mathcal{J})=x_5$.

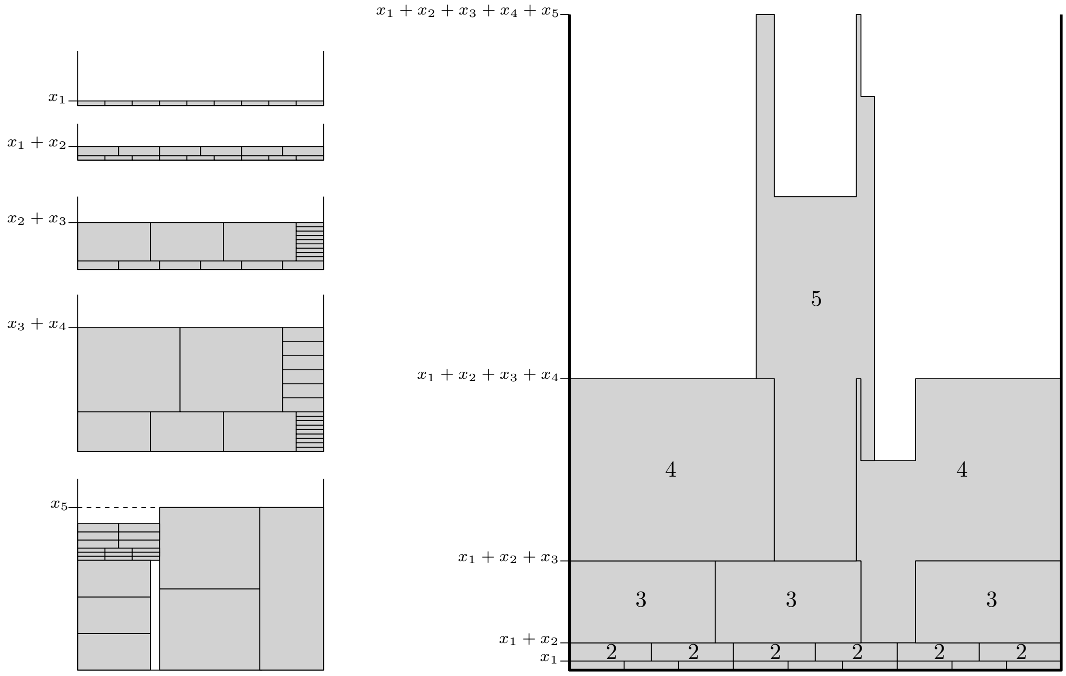


Figure 3 The lower bound of 1.812 for Online DSP. The indices of the jobs refer to the sets they are a part of; the jobs without index form the set $\mathcal{J}_1$. The optimal schedules of the sublists are depicted on the left and the final schedule of any algorithm that attempts to be strictly 1.812-competitive on the right.

Next, we consider the solutions constructed by the online algorithm. If the online algorithm ALG schedules the jobs in $\mathcal{J}_1$ such that at least two overlap, then its resulting peak is at least 2, while $\text{OPT}(\mathcal{J}_1) = 1$. Hence, ALG can be strictly 2-competitive at best:

$$\sup_{\mathcal{J}} \frac{\text{ALG}(\mathcal{J})}{\text{OPT}(\mathcal{J})} \geq \frac{\text{ALG}(\mathcal{J}_1)}{\text{OPT}(\mathcal{J}_1)} \geq 2 > \varrho.$$

Thus, if ALG schedules $\mathcal{J}_1$ without overlap, its load is x_1 in $[0, T]$. We now release $\mathcal{J}_2$. If at least two of the jobs in $\mathcal{J}_2$ overlap, ALG's peak is at least $x_1 + 2x_2$ compared to $\text{OPT}(\mathcal{J}_1\mathcal{J}_2) = x_1 + x_2$, which implies the first non-trivial bound on the competitive ratio

$$\sup_{\mathcal{J}} \frac{\text{ALG}(\mathcal{J})}{\text{OPT}(\mathcal{J})} \geq \frac{\text{ALG}(\mathcal{J}_1\mathcal{J}2)}{\text{OPT}(\mathcal{J}_1\mathcal{J}_2)} \geq \frac{x_1 + 2x_2}{x_1 + x_2} \geq \varrho.$$

Conversely, if ALG schedules $\mathcal{J}_2$ without overlap, its load is at least $x_1 + x_2$ in the time frame. We now release $\mathcal{J}_3$. If at least two $\mathcal{J}_3$ jobs overlap, ALG's peak is at least $x_1 + x_2 + 2x_3$ while $\text{OPT}(\mathcal{J}_1 \cdots \mathcal{J}_3) = x_2 + x_3$. Hence, we obtain

$$\sup_{\mathcal{J}} \frac{\text{ALG}(\mathcal{J})}{\text{OPT}(\mathcal{J})} \geq \frac{\text{ALG}(\mathcal{J}_1 \cdots \mathcal{J}_3)}{\text{OPT}(\mathcal{J}_1 \cdots \mathcal{J}_3)} \geq \frac{x_1 + x_2 + 2x_3}{x_2 + x_3} \geq \varrho.$$

If ALG schedules $\mathcal{J}_3$ without overlap, its load is at least $x_1 + x_2 + x_3$ for $3p_3 = 96$ units of time. Further, the only intervals where demand $x_1 + x_2$ is possible belong to $[0, 12) \cup [32, 44) \cup [64, 76) \cup [96, 108]$, and the length of such intervals is bounded from above by 12. The possible locations of these "gaps" can be seen in Figure 4.

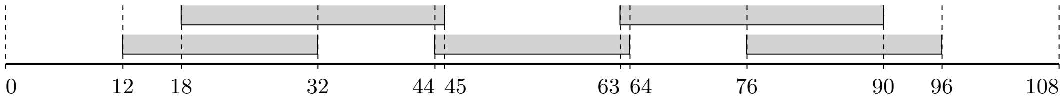


Figure 4 The intervals where ALG possibly has energy demand equal to x_1+x_2 after scheduling $\mathcal{J}_3$ are the white parts in the lower level, and the intervals where ALG possibly has a total load of $x_1+x_2+x_3$ after scheduling $\mathcal{J}_4$ are the white parts in the upper level.

We now release $\mathcal{J}_4$. Recall that $p_4 = 45$. Thus, the earliest end time of a job in $\mathcal{J}_4$ is 45 and the latest starting time is 63. Thus, if ALG schedules the two jobs in $\mathcal{J}_4$ to overlap, the interval where they overlap cannot completely belong to an interval with (prior) demand x_1+x_2. This implies that ALG's peak demand is at least $x_1+x_2+x_3+2x_4$, while $\textsc{Opt}(\mathcal{J}_1\cdots\mathcal{J}_4) = x_3+x_4$. Therefore,

$$\sup_{\mathcal{J}} \frac{\textsc{Alg}(\mathcal{J})}{\textsc{Opt}(\mathcal{J})} \geq \frac{\textsc{Alg}(\mathcal{J}_1\cdots\mathcal{J}_4)}{\textsc{Opt}(\mathcal{J}_1\cdots\mathcal{J}_4)} \geq \frac{x_1+x_2+x_3+2x_4}{x_3+x_4} \geq \varrho.$$

If ALG schedules $\mathcal{J}_4$ without overlap, its load is $x_1+x_2+x_3+x_4$ for a significant amount of time: the only intervals where no $\mathcal{J}_4$ job is scheduled belong to $[0,18)\cup[45,63)\cup[90,108]$. Combining with the possible gaps left by $\mathcal{J}_3$ jobs, ALG's schedule has an energy demand of $x_1+x_2+x_3+x_4$ in the intervals $[18,32)\cup[44,45)\cup[63,64)\cup[76,90)$, see Figure 4.

We now release the final job with $p_5 = 27$ and energy demand x_5. Hence, there will be a point $t\in[0,T]$ where ALG incurs a peak demand of at least $x_1+\cdots+x_5$, implying

$$\sup_{\mathcal{J}} \frac{\textsc{Alg}(\mathcal{J})}{\textsc{Opt}(\mathcal{J})} \geq \frac{\textsc{Alg}(\mathcal{J}_1\cdots\mathcal{J}_5)}{\textsc{Opt}(\mathcal{J}_1\cdots\mathcal{J}_5)} \geq \frac{x_1+x_2+x_3+x_4+x_5}{x_5} \geq \varrho.$$

Finally, we are interested in choosing $x_1,\ldots,x_5$ such that

$$\varrho = \min\left\{\frac{x_1+2x_2}{x_1+x_2}, \frac{x_1+x_2+2x_3}{x_2+x_3}, \frac{x_1+x_2+x_3+2x_4}{x_3+x_4}, \frac{x_1+x_2+x_3+x_4+x_5}{x_5}\right\}$$

is maximized while (C) and $x_i\geq 0$ for all $i\in[5]$ are satisfied. As can be seen from the Mathematica code that is available online[3], there exist x_i such that $\varrho\in(1.812,1.813)$. ◀

[3] `https://gitlab.com/bruchhold/online-demand-strip-packing`