Peak Demand Minimization via Sliced Strip Packing

Max A. Deppert
Universität Kiel, Germany

Klaus Jansen
Universität Kiel, Germany

Arindam Khan
Department of Computer Science and Automation, Indian Institute of Science, Bengaluru, India

Malin Rau
Universität Hamburg, Germany

Malte Tutas
Universität Kiel, Germany

Abstract

We study the Nonpreemptive Peak Demand Minimization (NPDM) problem, where we are given a set of jobs, specified by their processing times and energy requirements. The goal is to schedule all jobs within a fixed time period such that the peak load (the maximum total energy requirement at any time) is minimized. This problem has recently received significant attention due to its relevance in smart-grids. Theoretically, the problem is related to the classical strip packing problem (SP). In SP, a given set of axis-aligned rectangles must be packed into a fixed-width strip, such that the height of the strip is minimized. NPDM can be modeled as strip packing with slicing and stacking constraint: each rectangle may be cut vertically into multiple slices and the slices may be packed into the strip as individual pieces. The stacking constraint forbids solutions where two slices of the same rectangle are intersected by the same vertical line. Nonpreemption enforces the slices to be placed in contiguous horizontal locations (but may be placed at different vertical locations).

We obtain a $(5/3 + \varepsilon)$-approximation algorithm for the problem. We also provide an asymptotic efficient polynomial-time approximation scheme (AEPTAS) which generates a schedule for almost all jobs with energy consumption $(1 + \varepsilon)\mathrm{OPT}$. The remaining jobs fit into a thin container of height 1. The previous best result for NPDM was a 2.7 approximation based on FFDH [41]. One of our key ideas is providing several new lower bounds on the optimal solution of a geometric packing, which could be useful in other related problems. These lower bounds help us to obtain approximative solutions based on Steinberg's algorithm in many cases. In addition, we show how to split schedules generated by the AEPTAS into few segments and to rearrange the corresponding jobs to insert the thin container mentioned above.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases scheduling, peak demand minimization, approximation

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2021.21

Category APPROX

Related Version *Full Version*: `https://arxiv.org/abs/2105.07219`

Funding *Max A. Deppert*: Research supported by German Research Foundation (DFG) project JA 612/25-1.
Klaus Jansen: Research supported by German Research Foundation (DFG) project JA 612/25-1.
Malin Rau: Research supported by the French research program ENERGUMEN ANR-18-CE25-0008.

© Max A. Deppert, Klaus Jansen, Arindam Khan, Malin Rau, and Malte Tutas;
licensed under Creative Commons License CC-BY 4.0

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021).
Editors: Mary Wootters and Laura Sanità; Article No. 21; pp. 21:1–21:24
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Recent years have seen a substantial increase in the demand of electricity, due to rapid urbanization, economic growth and new modes of electrical energy consumption (e.g., electric cars). Traditionally, electricity generation, transmission, and distribution relied on building infrastructure to support the peak load, when the demand for electricity is maximum. However, the peak is rarely achieved and thus more demands can be accommodated using the inherent flexibility of scheduling of certain jobs. E.g., the energy requirements for HVAC units, electric vehicles, washers and dryers, water heaters, etc. can be met with a flexible scheduling of these appliances. Smart-grids [48, 32, 45] are next-generation cyber-physical systems that couple digital communication systems on top of the existing grid infrastructure for such efficient utilization of power, e.g., by shifting users' demand to off-peak hours in order to reduce peak load.

Future smart-grids are expected to obtain demand requirements for a time period and schedule the jobs such that the peak demand is minimized. Recently, this problem has received considerable attention [3, 37, 40, 42, 12]. Each job can also be modeled as a rectangle, with desired power demand as height and required running time as width. This gives a geometric optimization problem where the goal is to pack the slices of the rectangles into a strip of width as the time period. The goal is to minimize the maximum height of the packing. There is another additional *stacking constraint* requiring that no vertical line may intersect two slices from the same rectangle.

In this paper, we study this problem known as Nonpreemptive Peak Demand Minimization (NPDM)). Formally, we are given a set of jobs $\mathcal{J}$. Each job $j \in \mathcal{J}$ has a processing time $p(j) \in \mathbb{N}$ (also called width) and an energy requirement $e(j) \in \mathbb{N}$ (also called height). Furthermore, we are given a deadline $D \in \mathbb{N}$. All the jobs are available from the time 0 and have to be finished before the deadline D. A schedule σ of the jobs $\mathcal{J}$ assigns each job a starting time $\sigma(j) \in \mathbb{N}$ such that it is finished before the deadline, i.e., $c(j) := \sigma(j) + p(j) \leq D$. The total energy consumption at a time $\tau \in \{0, \dots, D\}$ is given by $e(\tau) := \sum_{j \in \mathcal{J}, \sigma(j) \leq \tau < \sigma(j)+p(j)} e(j)$. The objective is to minimize the peak of energy consumption, i.e., minimize $T_\sigma := \max_{\tau \in \{0,\dots,D-1\}} e(\tau)$.

NPDM can be viewed as a variant of strip packing problem, where we are allowed to slice the rectangles vertically and the slices must be packed in contiguous horizontal positions (but may be placed at different vertical positions). In the classical strip packing problem, we are given a set of rectangles as well as a bounded-width strip and the objective is to find a non-overlapping, axis-aligned packing of all rectangles into the strip so as to minimize the height of the packing. A simple reduction from the PARTITION problem shows a lower bound of 3/2 for polynomial-time approximation for the problem. In 1980, Baker et al. [4] first gave a 3-approximation algorithm. Later Coffman et al. [31] introduced two simple shelf-based algorithms: Next Fit Decreasing Height (NFDH), First Fit Decreasing Height (FFDH), with approximation ratios as 3 and 2.7, respectively. Sleator [46] gave a 2.5-approximation. Thereafter, Steinberg [47] and Schiermeyer [44] independently improved the approximation ratio to 2. Afterwards, Harren and van Stee [21] obtained a 1.936-approximation. The present best approximation is $(5/3 + \varepsilon)$, due to Harren et al. [20].

Alamdari et al. [3] studied a variant where we allow preemption of jobs, also known as two-dimensional strip packing with slicing and stacking constraints (2SP-SSC), or preemptive offline cost optimal scheduling problem (P-OCOSP) [41]. They showed this variant to be NP-hard and obtained an FPTAS. They also studied several shelf-based algorithms and provide a practical polynomial time algorithm that allows only one preemption per job. Ranjan et al. [42] have proposed a practical 4/3-approximation algorithm for this problem.

For NPDM, Tang et al. [48] first proposed a 7-approximation algorithm. Yaw et al. [49] showed that NPDM is NP-hard to approximate within a factor better than 3/2. They have given a 4-approximation for a special case when all jobs require the same execution time. Ranjan et al. [40], have proposed a 3-approximation algorithms for NPDM. They [41] also proposed an FFDH-based 2.7-approximation algorithm for a mixed variant where some jobs can be preempted and some can not be preempted.

Our Contributions. We obtain improved approximation algorithms for NPDM.[1] Note that the optimal solutions of sliced strip packing/NPDM and strip packing can be quite different. In fact, in [9] an example with a ratio 5/4 is presented. Thus, the techniques from strip packing do not always translate directly to our problem. We exploit the property that, due to slicing, we can separately guess regions (*profile*) for packing of jobs with large energy demand (*tall* jobs) and jobs with large time requirements (*wide* jobs). We show that we can remove a small amount of jobs with large energy demand so that we can approximately guess the optimal profile of jobs with large processing time so that their starting positions come from a set containing a constant number of values. This helps us to show the existence of a structured solution that we can pack near-optimally using linear programs. This shows the existence of an asymptotic efficient polynomial-time approximation scheme (AEPTAS):

▶ **Theorem 1.** *For any $\varepsilon > 0$, there is an algorithm that schedules all jobs such that the peak load is bounded by $(1+\varepsilon)\mathrm{OPT}+e_{\max}$, where $e_{\max}$ denotes the maximal energy demand among the given jobs. The time complexity of this algorithm is bounded by $\mathcal{O}(n\log(n)) + 1/\varepsilon^{1/\varepsilon^{\mathcal{O}(1/\varepsilon)}}$.*

In fact, we show a slightly stronger result here, providing a schedule for almost all jobs $\mathcal{J}\setminus C$ with peak energy demand bounded by $(1+\varepsilon)\mathrm{OPT}$ plus a schedule for the remaining jobs C with peak energy demand $e_{\max}$ and schedule length λD for a sufficiently small $\lambda \in [0,1]$.

Using the AEPTAS and Steinberg's algorithm[47], we obtain our main result:

▶ **Theorem 2.** *For any $\varepsilon > 0$, there is a polynomial-time $(5/3+\varepsilon)$-approximation algorithm for NPDM.*

Previously, in strip packing (and related problems) the lower bound on the optimal packing height is given based on the height of the tallest job or the total area of all jobs [47, 44]. One of our main technical contributions is to show several additional lower bounds on the optimal load. These bounds may be helpful in other related geometric problems. In fact, these can be helpful to simplify some of the analyses of previous algorithms. Using these lower bounds, we show, intuitively, that if there is a large amount of energy consuming jobs (or time consuming jobs) we can obtain a good packing using Steinberg's algorithm. Otherwise, we start with the packing from AEPTAS and modify the packing to obtain a packing within $(5/3+\varepsilon)$-factor of the optimal. This repacking utilizes novel insights about the structure of the packing that precedes it, leading to a less granular approach when repacking.

Related Work. Strip packing has also been studied under asymptotic approximation. The seminal work of Kenyon and Rémila [34] provided an APTAS with an additive term $O(e_{\max}/\varepsilon^2)$, where $e_{\max}$ is the height of the tallest rectangle. The latter additive term was subsequently improved to $e_{\max}$ by Jansen and Solis-Oba [28]. Pseudo-polynomial time algorithm for strip packing has received recent attention [39, 18, 2, 22]. Finally, Jansen and

[1] The same result as in Theorem 1 was achieved independently in [15]; their approach is however substantially different from ours.

Rau [27] gave an almost tight pseudo-polynomial time $(5/4+\varepsilon)$-approximation algorithm. Recently, Galvez et al. [14] gave a tight $(3/2+\varepsilon)$-approximation algorithm for a special case when all rectangles are skewed (each has either width or height $\leq \delta D$, where $\delta \in (0,1]$ is a small constant).

A related problem is non-contiguous multiple organization packing [10], where the width of each rectangle represents a demand for a number of concurrent processors. This is similar to sliced strip packing, however, the slices need to be horizontally aligned to satisfy concurrency. Several important scheduling problems are related, such as multiple strip packing [27], malleable task scheduling [23], parallel task scheduling [29], moldable task scheduling [24, 25], etc.

Several geometric packing problems are well-studied in combinatorial optimization. In two-dimensional bin packing, we are given a set of rectangles and the goal is to pack all rectangles into the minimum number of unit square bins. This well-studied problem [5, 26] is known to admit no APTAS [6], unless P=NP, and the present best approximation ratio is 1.406 [7]. Another related problem is two-dimensional geometric knapsack [30, 17], where each rectangle has an associated profit and we wish to pack a maximum profit subset of rectangles in a given square knapsack. The present best approximation ratio for the problem is 1.89 [16]. These problems are also studied under guillotine cuts [8, 36, 35] where all jobs can be cut out by a recursive sequence of end-to-end cuts. There are several other important related problems such as maximum independent set of rectangles [1], unsplittable flow on a path [19], storage allocation problem [38], etc. We refer the readers to [13] for a survey.

The following result from [47] will be a crucial subroutine in our algorithms.

▶ **Theorem 3** (Steinberg's Algorithm [47]). *Steinberg's algorithm packs a set of rectangular objects $\mathcal{R}$ into a rectangular container of height a and width b in polynomial time, if and only if the following inequalities hold:*

$$e_{\max} \leq a, \quad p_{\max} \leq b, \quad 2\sum_{r\in\mathcal{R}} e(r)p(r) \leq ab-(2e_{\max}-a)_+(2p_{\max}-b)_+, \quad \text{(Steinberg Cond.)}$$

where $x_+ = \max\{x,0\}$, $p_{\max}$ is the maximal width of a rectangle, and $e_{\max}$ is the maximal height of a rectangle, $e(r)$ represents the height of a rectangle and $p(r)$ represents the width of a rectangle.

General Approach. The general idea of our $(5/3+\varepsilon)$-approximation algorithm is as follows: If the jobs that are large in at least one of the two dimensions have a sufficiently large total amount of work, a $(5/3+\varepsilon)$-approximation can be achieved by placing these jobs in a structured manner and using Steinberg's algorithms to place the residual jobs. We describe two of these cases in Section 2.

Otherwise, we know that the total amount of work of these jobs not too large. In this case, we find a schedule σ_1 that schedules almost all the jobs using an energy demand of at most $(1+\mathcal{O}(\gamma))$OPT. The residual jobs are contained in an extra schedule σ_2 of length γD and peak energy demand bounded by T. These schedules are generated using the algorithm described Section 4 in the proof of Theorem 10. Note that we have to choose $\gamma \in \mathcal{O}_\varepsilon(1)$ small enough, in order to meet the requirements for the next step.

Given these schedules, we find a rescheduling argument, where we rearrange the schedule σ_1 such that we can add the schedule σ_2 while increasing the peak energy demand by at most $(2/3)T$. This repacking argument is described in Section 3 in the proof of Theorem 7.

Notation. For an instance $I = (\mathcal{J}, D)$, we denote by $\mathrm{OPT}(I)$ (or just OPT) the optimal energy consumption peak. For some set of jobs $\mathcal{J}$ we define $\mathrm{work}(\mathcal{J}) = \sum_{i\in\mathcal{J}} p(i)e(i)$, the total processing time as $p(\mathcal{J}) = \sum_{i\in\mathcal{J}} p(i)$, as well as the total energy demand as $e(\mathcal{J}) = \sum_{i\in\mathcal{J}} e(i)$. With the additional notation of $\mathcal{J}_{P(i)} = \{ i \in \mathcal{J} \mid P(i) \}$ as a restriction of $\mathcal{J}$ using the predicate P. E.g., we may express the energy demand of jobs of $\mathcal{J}$ which have a processing time of at least $D/2$ by $e(\mathcal{J}_{p(i)\geq D/2})$. Furthermore, given a set of jobs $\mathcal{J}$, we denote $p_{\max}(\mathcal{J}) := \max_{i\in\mathcal{J}} p(i)$ and $e_{\max}(\mathcal{J}) := \max_{i\in\mathcal{J}} e(i)$ and write $e_{\max}$ and $p_{\max}$ if the set of jobs is clear from the context. We say a job i that is placed at $\sigma(i)$ overlaps a point in time τ if and only if $\sigma(i) \leq \tau < \sigma(i) + p(i)$. The set $\mathcal{J}(\tau)$ denotes the set of jobs that overlap the point in time τ. Additionally, we introduce segments S of the schedule which refer to time intervals and container C which can be seen as sub schedules. The starting point of a time interval S will be denoted by $\sigma(S)$ and its endpoint as $c(S)$. On the other hand, a container C has a length (time), which is denoted as $p(C)$, and a bound on the energy demand $e(C)$. If these containers are scheduled, the get a start point $\sigma(C)$, which is added to the start point of any job scheduled in C.

2 Cases solved with Steinberg's algorithm

In this section, we first bound the peak energy demand from below and then use Steinberg's algorithm to handle some cases. Two obvious lower bounds are the energy demand of the most energy demanding job $e_{\max}$ and the bound given by the total amount of work of the jobs, i.e., $\mathrm{OPT} \geq \max\{e_{\max}, \mathrm{work}(\mathcal{J})/D\}$. Another simple lower bound is the total energy demand of jobs longer than $D/2$, since they have to be scheduled in parallel. This gives us the first lower bound on OPT and we call it $T_1 := \max\{e_{\max}, \mathrm{work}(\mathcal{J})/D, e(\mathcal{J}_{p(i)>D/2})\}$. In the following, we will present three more complex lower bounds. The next bound is related to the items with a large energy demand. We denote this lower bound as

$$T_2 := \min\{T | p(\mathcal{J}_{e(i)\geq T/3}) + p(\mathcal{J}_{e(i)\geq 2T/3}) \leq 2D \wedge p(\mathcal{J}_{e(i)\geq T/2}) \leq D\}.$$

The next two lower bounds depend on the ratio of long jobs and jobs with large energy demand. For a given $k \in [n]$, we define $\mathcal{J}_k$ to be the set of the k jobs with the largest energy demand in $\mathcal{J}$ and $\mathcal{J}'_k$ to be the set of the k jobs with the largest energy demand in $\mathcal{J} \setminus \mathcal{J}_{p(i)>D/2}$. Let i_k and i'_k be the jobs with the smallest energy demand in $\mathcal{J}_k$ and $\mathcal{J}'_k$, respectively. We define:

$$T_{3,a} := \max\{\min\{e(i_k) + e(\mathcal{J}_{p(i)>D-p(\mathcal{J}_k)/2} \setminus \mathcal{J}_k), 2e(i_k)\} | k \in \{1,\dots,n\}, p(\mathcal{J}_k) \leq D\},$$
$$T_{3,b} := \max\{\min\{e(i'_k) + e(\mathcal{J}_{p(i)>D-p(\mathcal{J}'_k)/2}), 2e(i'_k)\} | k \in \{1,\dots,n\}, p(\mathcal{J}'_k) \leq D\},$$

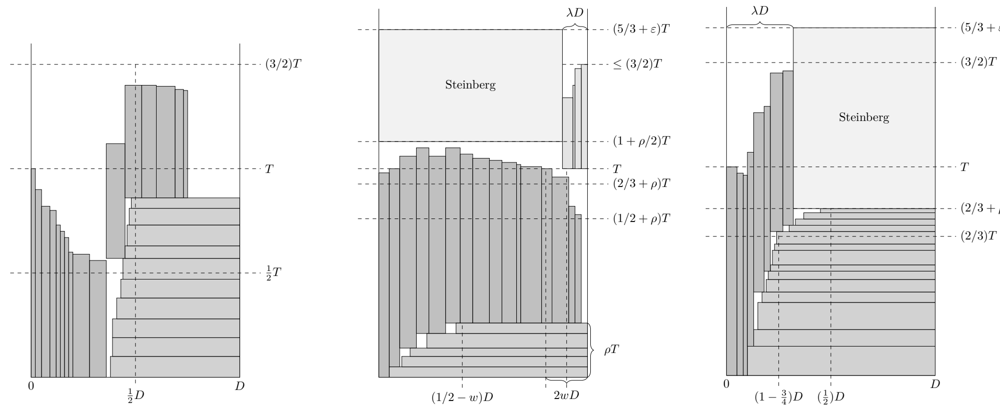
and $T_3 = \max\{T_{3,a}, T_{3,b}\}$.

Finally, define $\mathcal{J}_k$ as the set of the k jobs with largest energy demand, $\mathcal{J}_{D,k} := \mathcal{J}_{p(i)>(\max\{D-p(\mathcal{J}_k),D/2\})} \setminus \mathcal{J}_k$. Let i_k be the job with the smallest energy demand in $\mathcal{J}_k$, then define

$$T_4 := \max\{\min\{2e(i_k), e(i_k) + e(\mathcal{J}_{D,k})/2\} | k \in \{1,\dots,n\}, p(\mathcal{J}_k) \leq D\}.$$

▶ **Theorem 4.** *Given an instance $I = (\mathcal{J}, D)$, the value $T := \max\{T_1, T_2, T_3, T_4\}$ is a lower bound on $\mathrm{OPT}(I)$. The value of T can be found in $\mathcal{O}(n \log(n))$.*

This lower bound on OPT helps us to identify two cases, that can be solved by scheduling jobs that are large in at least one of the two dimensions in a sorted manner, while the other jobs are scheduled using Steinberg's algorithm. We prove this and the following two theorems in the appendix.



(a) An L-shaped schedule. **(b)** First Steinberg case. **(c)** Second Steinberg case.

Figure 1 Subfigure **1a** shows one possible L-shaped schedule, where $\mathcal{J}_{\text{seq}}$ contains all the jobs with energy demand larger than $T/2$ and $\mathcal{J}_D$ contains all the jobs with processing time larger than $D/2$. Subfigure **1b** shows a schedule in the case that $p(\mathcal{J}_{e(i)>(2/3)T}) \geq (1-w)D$. Subfigure **1c** shows a schedule in the case that $e(\mathcal{J}_{p(i)>(3/4)D}) \geq (2/3)T$.

▶ **Theorem 5** (First Steinberg Case)**.** *Let $T := \max\{T_1, T_2, T_3, T_4\}$ be the lower bound on* OPT *as defined above. If $p(\mathcal{J}_{e(j)>(2/3)T}) \geq (1-(3/4)\varepsilon)D$, there is a polynomial time algorithm to place all jobs inside a schedule with peak energy demand at most $(5/3+\varepsilon)T$.*

▶ **Theorem 6** (Second Steinberg Case)**.** *Let T be the lower bound on* OPT *defined as above. If $e(\mathcal{J}_{p(i)\geq(3/4)D}) > (2/3)T$, then there is a polynomial time algorithm that places all the jobs inside the area $[0, D] \times [0, (5/3)T]$.*

3 $(5/3+\varepsilon)$-Approximation

This section inspects schedules generated by the AEPTAS from Theorem 10, more closely. The AEPTAS generates a schedule that fits almost all jobs into an amount of work of peak energy demand $T \leq (1+\varepsilon)\text{OPT}$. Left out of the schedule is a set of jobs that has a very small total processing time, where each job can have an energy demand up to OPT. As such, this set of jobs can be fit into a strip of energy demand OPT and processing time γD for some $\gamma > 0$. Since we aim to generate a schedule of peak energy demand $(5/3+\varepsilon)\text{OPT}$, it does not suffice to simply place this set atop the generated schedule, as this would result in a peak energy demand of 2OPT. Instead, we must find some area in the generated schedule, inside of which we can remove jobs such that an energy demand of OPT/3 for a processing time of λD is empty, where $\lambda \in [0,1]$ is a small constant depending on ε. Once we have achieved this, and placed the jobs removed by this procedure in a way that does not intersect this strip, we can then place the strip of energy demand OPT at exactly that place, resulting in a schedule of peak energy demand $(5/3+\varepsilon)\text{OPT}$. If none of the previously mentioned cases (as in Theorem 5 and 6, that can be solved using Steinberg's algorithm) apply, then the following Theorem combined with Theorem 10 proves Theorem 2.

▶ **Theorem 7.** *Let $\varepsilon \in (0, 1/3]$, $\varepsilon' \leq (3/5)\varepsilon$ and $\gamma \leq (3/40)\varepsilon$. Given an instance I with $e(\mathcal{J}_{p(j)>(3/4)D}) \leq (2/3)T'$ and $p(\mathcal{J}_{e(j)>(2/3)T'}) \leq (1-(3/4)\varepsilon)D$, for $T' = \max\{T_1, T_2, T_3, T_4\}$ and a schedule σ (e.g. generated by the APTAS) where almost all jobs are placed such that the peak energy demand is $T \leq (1+\varepsilon')\text{OPT}$, and the residual jobs inside an additional box C_γ of energy demand T and processing time γD, we can find a restructured schedule that places all the jobs up to a schedule with peak energy demand of at most $(5/3+\varepsilon)\text{OPT}$.*

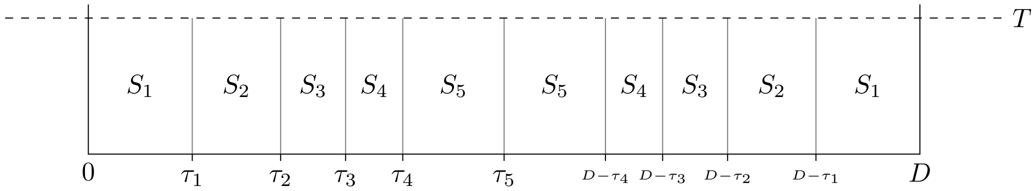


Figure 2 Splitting the given schedule into segments at time points $\tau_1 = \frac{D}{8}$, $\tau_2 = \frac{(15-24\gamma)D}{64}$, $\tau_3 = \frac{(9+11\gamma)D}{32}$, $\tau_4 = \frac{(3+2\gamma)D}{8}$ and $\tau_5 = \frac{D}{2}$.

Proof. From the schedule σ, we will generate a new schedule σ'. Some jobs will be shifted to new starting positions σ'. Other jobs j that are not mentioned in this proof keep their original starting positions, i.e., $\sigma'(j) = \sigma(j)$.

If the schedule contains a job j with processing time $p(j) \in [\gamma, (1-2\gamma)D]$ and energy demand $e(j) \in [(1/3)T, (2/3)T]$, we proceed as follows to make room for C_γ by shifting job j: Since $p(j) \leq (1-2\gamma)D$ it holds that $\max\{\sigma(j), D - (\sigma(j) + p(j))\} \geq \gamma D$. Let us, w.l.o.g., assume that $\sigma(j) \leq D - (\sigma(j) + p(j))$, otherwise we mirror the schedule at $D/2$. We shift the job j completely to the right (by at least γD) such that it is positioned at $\sigma'(j) := D - p(j)$. This increases the peak energy demand to at most $(5/3)T$. Now the schedule between $\sigma(j)$ and $\sigma(j) + \gamma D$ has an energy demand of at most $(2/3)T$. We place the box C_γ at $\sigma(j)$. Since the box has an energy demand of at most T, the resulting schedule still has a peak energy demand of at most $(5/3)T \leq (5/3 + \varepsilon)\mathrm{OPT}$.

If such a job does not exist, we search for segments that are not overlapped by jobs with energy demand larger than $(2/3)T$. We split the schedule into segments at the times $\tau_1 = \frac{D}{8}$, $\tau_2 = \frac{(15-24\gamma)D}{64}$, $\tau_3 = \frac{(9+11\gamma)D}{32}$, $\tau_4 = \frac{(3+2\gamma)D}{8}$ and $\tau_5 = \frac{D}{2}$ as well at $\tau_i' = D - \tau_i$ for $i \in \{1,2,3,4\}$ and set $\tau_0 = 0$. We number the resulting segments in increasing order from 0 to $D/2$ and from D to $D/2$ such that similar segments on both sides get the same number, see Figure 2. For $k \in \{1,2,3,4,5\}$, we denote by $\sigma(S_k)$ $(= \tau_{k-1})$ the start-time of a segment, by $c(S_k)$ $(= \tau_k)$ the end-time of the segment and by $p(S_k)$ $(= \tau_k - \tau_{k-1})$ the processing time of the segment S_k . Since $p(\mathcal{J}_{e(j)>(2/3)T}) \leq (1-(3/4)\varepsilon)D$, we know, by pigeonhole principle, that in one of these segments a total time of at least $(3/4)\varepsilon D/10 \geq (3/40)\varepsilon D \geq \gamma D$ is not overlapped by these jobs.

Let S_{l,k_1} be the earliest such strip, and S_{r,k_2} the latest (they might be the same) such that $k_1, k_2 \in \{1,2,3,4,5\}$ represent the index of the strips $S_1, \dots, S_5$. In the next step, we modify the start- and end-times of S_{l,k_1} such that it starts at the end of a job with energy demand at least $(2/3)T$ or at 0. We denote the shifted start times as $\sigma'(\cdot)$ and the shifted completion time as $c'(\cdot)$. If the start-time of S_{l,k_1}, i.e. τ_{k-1} intersects a job j with $e(j) \geq (2/3)T$, we define $\sigma'(S_{l,k_1}) := \sigma(j) + p(j) \leq c(S_{l,k_1}) - \gamma D$. Otherwise if $k_1 \neq 1$, we find the last job j ending before $\sigma(S_{l,k_1})$ with $e(j) \geq (2/3)T$ and define $\sigma'(S_{l,k_1}) := \sigma(j) + p(j) \geq \sigma(S_{l,k_1}) - \gamma D$ and shift the end-time of S_{l,k_1} by the same amount. Note that, since S_{l,k_1} is the first strip with at least γD time not occupied by jobs with energy demand larger than $(2/3)T$, the starting time of S_{l,k_1} is reduced by at most γD, while the processing time of the segment is not increased. Finally, if the end-time of S_{l,k_1} intersects a job j that has an energy demand larger than $(2/3)T$, we reduce it to $c'(S_{l,k_1}) := \sigma(j)$ and call the modified segment S'_{l,k_1}. These modifications never decrease the total time that is not overlapped by jobs with energy demand larger than $(2/3)T$ in S_{l,k_1}. We do the same but mirrored for S_{r,k_2} resulting in a modified segment S'_{r,k_2}. For an illustration of this procedure, see Figure 3. If $D - c(S'_{r,k_2}) \leq \sigma(S'_{l,k_1})$, we mirror the schedule such that $\sigma'(j) = D - c(j)$. We denote by S'_k

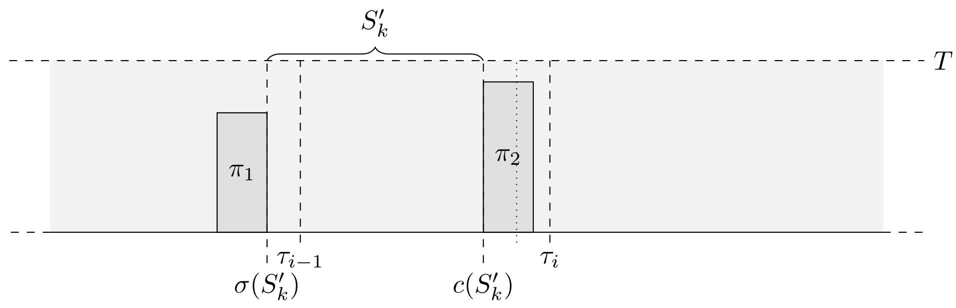


Figure 3 An illustration of the border shifting procedure. The original borders are indicated by τ_{i-1} and τ_i. As τ_{i-1} is not intersected by a job with energy demand larger than $(2/3)T$ we shift $\sigma(S'_k)$ to an earlier point in time, such that the job with energy demand larger than $(2/3)T$ π_1 ends at the exact same time. We then shift $c(S'_k)$ by the same amount. The shifted $c(S'_k)$ may intersect a job with energy demand larger than $(2/3)T$ π_2, indicated by the dotted line, and in this case we shift the border further such that $c(S'_k) = \sigma(\pi_2)$ holds.

the segment in $\{S'_{l,k_1}, S'_{r,k_2}\}$ that appears first in this new schedule, where $k = \min\{k_1, k_2\}$ represents the original number of the chosen segment. As a consequence of this mirroring if $k \geq 2$, we ensured there exists a job j with $e(j) > (2/3)T$ and $c(S'_k) \leq \sigma(j) \leq D - \sigma(S'_k)$. Additionally, we know about the start and endpoints of this segment that $\tau_{k-1} - \gamma D \leq \sigma(S'_k)$ and $p(S'_k) \leq \tau_k - \tau_{k-1}$.

We aim to remove jobs from S'_k, such that the peak energy consumption reached inside S'_k is bounded by $(2/3)T$. We categorize the jobs to be removed in three classes; first the set of jobs $\mathcal{J}_{\text{cont}}$ that are wholly contained in S'_k due to the earlier shifting and have an energy demand less than $(2/3)T$, second the set of jobs that have an energy demand larger than $(2/3)T$, and finally the set of jobs intersecting one of the time points $\sigma(S'_k)$ or $c(S'_k)$. First, we remove $\mathcal{J}_{\text{cont}}$ from the segment and schedule them inside a container that has an energy demand of at most $(2/3)T$ and length at most $3p(S'_k)$.

▶ **Lemma 8.** *The jobs $\mathcal{J}_{\text{cont}}$ can be scheduled inside a container C_{cont} of energy demand $(2/3)T$ and processing time $3p(S'_k) \leq D/2$.*

Proof. First note that $\text{work}(\mathcal{J}_{\text{cont}}) \leq p(S'_k)T$, since the peak energy demand in σ is bounded by T. We place these jobs using Steinberg's algorithm. Recall that this procedure allows us to place a set of rectangles $\mathcal{R}$ into a container of size $a \cdot b$ as long as the following conditions are met: $p_{max}(\mathcal{J}) \leq a, e_{\max}(\mathcal{J}) \leq b, 2 \cdot \text{work}(\mathcal{J}) \leq (ab - (2e_{\max}(\mathcal{J}) - T)_+(2p_{\max}(\mathcal{J}) - D)_+)$. Setting our values for $b = 3p(S'_k)$ and $a = (2/3)T$ yields the desired property. Clearly no job wholly contained in a segment of processing time $p(S'_k)$ can have a processing time greater than $p(S'_k)$. Furthermore, the maximum energy demand of any job in $\mathcal{J}_{\text{cont}}$ is $(2/3)T$. Finally, we have:

$$
\begin{aligned}
2 \cdot \text{work}(\mathcal{J}_{\text{cont}}) &\leq 3p(S'_k) \cdot (2/3)T - (2p(S'_k) - 3p(S'_k))_+ \cdot (2(2/3)T - (2/3)T)_+ \\
&= (ab - (2e_{\max}(\mathcal{J}) - b)_+(2p_{\max}(\mathcal{J}) - a)_+)
\end{aligned}
$$
◀

In the next step, we consider the jobs with energy demand larger than $(2/3)T$. By construction of the strip, we know that the total processing time of jobs with energy demands larger than $(2/3)T$ is bounded by $p(S'_k) - \gamma D$. We remove all these jobs from the strip and combine them with the extra container C_γ of energy demand at most T and processing time γD to a new container called C_{tall}. It has an energy demand of at most T and a processing time of at most $p(S'_k)$.

After this step, the only jobs remaining inside the area of S'_k are the jobs that overlap the borders of S'_k. If the peak energy demand in S'_k is lower than $(2/3)T$, we place the container C_{tall} inside the strip S'_k as well as the container C_{cont} right of $D/2$ and are done. Otherwise, we have to remove jobs that overlap the borders of the strip S'_k until the peak energy demand in S'_k is bounded by $(2/3)T$. The jobs we choose to remove are dependent on the position of the strip. The following lemma helps to see how these jobs can be shifted without increasing the peak energy demand of the schedule too much.

▶ **Lemma 9.** *Consider a schedule σ with peak energy demand bounded by T and a time $\tilde{\tau}$, as well as a subset of jobs $\mathcal{J}_{\text{Move}} \subseteq \mathcal{J}(\tilde{\tau})$ with $e(\mathcal{J}_{\text{Move}}) \leq a \cdot T$ for some $a \in [0,1]$. Let τ be the smallest value $\sigma(j)$ for $j \in \mathcal{J}_{\text{Move}}$. Consider the schedule σ', where all the jobs in $\mathcal{J}_{\text{Move}}$ are delayed such that they end at D, i. e., $\sigma'(j) = D - p(j)$ for all $\mathcal{J}_{\text{Move}}$ and $\sigma'(j) = \sigma(j)$ for all other jobs.*

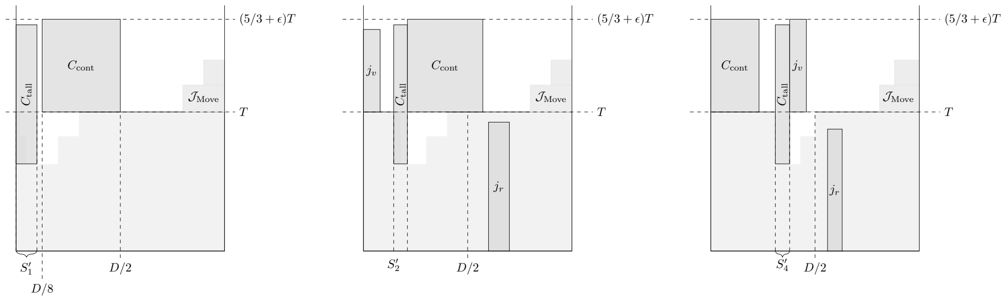
In the schedule σ' before of $D/2 + \tau/2$, the peak energy demand is bounded by T, while after of $D/2 + \tau/2$ the peak energy demand is bounded by $(1 + a)T$.

Proof. If the energy demand of the schedule σ' is larger than T at a position τ', it has to be because one of the jobs in $\mathcal{J}_{\text{Move}}$ overlaps it. Hence, that peak energy demand is bounded by $(1 + a)T$, since we shifted jobs with total energy demand bounded by aT. Let j be one of the shifted jobs. If $\sigma'(j) > D/2 + \tau/2$, the energy demand of the schedule before of $D/2 + \tau/2$ cannot be influenced by this job. Therefore, assume that $D - p(j) = \sigma'(j) \leq D/2 + \tau/2$. As a consequence, $p(j) \geq D/2 - \tau/2$. Since $\sigma(j) \leq \tau$ it holds that $\sigma(j) + p(j) \geq D/2 + \tau/2$. Thus before time $D/2 + \tau/2$, the job j overlaps its previous positions and cannot increase the peak energy demand above T. ◀

We choose which of the overlapping jobs to shift depending if $k = 1$ or $k \neq 1$. Remember that none of the borders of S'_k overlap a job that has an energy demand larger than $(2/3)T$, and assume for the following, that there is a point inside S'_k where the total energy demand of overlapping jobs is larger than $(2/3)T$.

Case 1: $k = 1$. Consider the time $\tau = c(S'_1) \leq D/8$ and the set of jobs $\mathcal{J}(\tau)$ that are intersected by this line. We know that the total energy demand of jobs with processing time greater than $(3/4)D$ is bounded by $(2/3)T$. Let $\mathcal{J}_{\text{Move}}$ be the set of jobs generated as follows: Greedily take the jobs with the largest energy demand from $\mathcal{J}(\tau) \setminus \mathcal{J}_{p(j)>(3/4)D}$, until either all the jobs from $\mathcal{J}(\tau) \setminus \mathcal{J}_{p(j)>(3/4)D}$ are contained in $\mathcal{J}_{\text{Move}}$ or $e(\mathcal{J}_{\text{Move}}) \in [(1/3)T, (2/3)T]$. In this process, we never exceed $(2/3)T$ since, if there is a job with energy demand larger than $(1/3)T$ in $\mathcal{J}(\tau) \setminus \mathcal{J}_{p(j)>(3/4)D}$, we choose it first and immediately stop. We delay the jobs in $\mathcal{J}_{\text{Move}}$ to new start positions σ' such that for each job $j \in \mathcal{J}_{\text{Move}}$ we have that $c'(j) := \sigma'(j) + p(j) = D$. Note that $\sigma'(j) \geq (1/4)D$ for each $j \in \mathcal{J}_{\text{Move}}$ and therefore no longer overlaps $c(S'_1)$. Furthermore, we know by Lemma 9 that before $D/2$ the peak energy demand is bounded by T, while after $D/2$ the peak energy demand is bounded by $(5/3)T$. Furthermore, the peak energy demand inside S'_1 is bounded by $(2/3)T$.

Since S'_1 has a processing time of at most $D/8$, we know by Lemma 8 that $\mathcal{J}_{\text{cont}}$ can be placed inside a container C_{cont} with energy demand at most $(2/3)T$ and processing time bounded by $3D/8$. Therefore, we can schedule this container at $D/8$ and know that it is finished before $D/2$. Finally, we schedule the container C_{tall} at $\sigma'(C_{\text{tall}}) = 0$. The peak energy demand of the resulting schedule is bounded by $(5/3)T$. See Figure 4a for the repacking procedure.



(a) Repacking for $k = 1$. **(b)** Repacking for $k \in \{2, 3\}$. **(c)** Repacking for $k \in \{4, 5\}$.

Figure 4 Illustration of the steps in the proof of Theorem 7. Note that the set $\mathcal{J}_{\text{Move}}$ is delayed such that the jobs end at D. The containers C_{tall} and C_{cont} are placed such that they do not intersect. For 4b and 4c, the jobs j_v are placed in the same manner, and the job j_r is denoted.

Case 2: $k \neq 1$. In this case, the borders of the considered strip can be overlapped from both sides. Furthermore, we know that the left border of S'_k is right of $D/8 - \gamma D \geq \gamma D$.

Consider the largest total energy demand of jobs that are intersected by any vertical line through S'_k and denote this energy demand as $T_{S'_k}$. Since there is a job j_l with energy demand larger than $(2/3)T$ with $\sigma(j_l) + p(j_l) = \sigma(S'_k)$, we know that the total energy demand of jobs intersecting $\sigma(S'_k)$ can be at most $(1/3)T$. Next, consider the closest job j_r that starts after $c(S'_k)$ and has an energy demand larger than $(2/3)T$. By the choice of S'_k, we know that such a job must exist and that $\sigma(j_r) \leq D - \sigma(S'_k)$, by the choice of S'_k out of $S'_{k_1,l}$ and $S'_{k_2,r}$. Furthermore, we know that the total energy demand of jobs intersecting the vertical line at $\sigma(j_r)$ is bounded by $(1/3)T$.

Hence the jobs that overlap the vertical line at $\sigma(j_r)$ and the jobs that overlap the vertical line at $\sigma(S'_k)$ add a total energy demand of at most $(2/3)T$ to $T_{S'_k}$. Let us now consider the jobs $\mathcal{J}_M$ that overlap the time $c(S'_k)$ but neither the time $\sigma(S'_k)$ nor the time $\sigma(j_r)$. Each of them has a processing time of at most $D - \sigma(S'_k) - \sigma(j_r) \leq D - 2\sigma(S'_k)$. Hence when delaying their start points such that $\sigma'(j) = D - p(j)$, they no longer overlap the time $c(S'_k)$ since $p(S'_k) \leq \sigma(S'_k)$ for each $k \in \{2, 3, 4, 5\}$.

We greedily take jobs from $\mathcal{J}_M$ that have the earliest starting point until we have all jobs from $\mathcal{J}_M$ or we have a total energy demand of at least $(1/3)T$. If the total energy demand of the chosen jobs is larger than $(2/3)T$, the last job j_v has an energy demand of at least $(1/3)T$. Since it has a processing time lower than $(1 - 2\gamma)D$, it has to have a processing time of at most γD. We remove this job and place it later, while we shift all the others to new positions σ' such that $\sigma'(j) = D - p(j)$ for each of the taken jobs j. We call the set of shifted jobs $\mathcal{J}_{\text{Move}}$.

Furthermore, since $\mathcal{J}_{\text{Move}}$ has a total energy demand of at most $(2/3)T$ and a starting point right of $\sigma(S'_k)$, we know by Lemma 9 that the peak energy demand right of $D/2 + \sigma(S'_k)/2$ is bounded by $(5/3)T$ while left of $D/2 + \sigma(S'_k)/2$ it is bounded by T.

Let $\sigma(j_l)$ be the starting time of the last taken job. Before $\sigma(j_l)$ (in S'_k) there is no longer a job from $\mathcal{J}_M$, and, hence inside in the strip S'_i that is left of $\sigma(j_l)$, the peak energy demand is bounded by $(2/3)T$. On the other hand, after $\sigma(j_l)$ (in S'_k) we either have removed jobs with total energy demand at least $(1/3)T$, or all the jobs from $\mathcal{J}_M$ and hence the schedule there can have a total energy demand of at most $(2/3)T$ as well. Therefore, we can place the container C_{tall} inside S'_k without increasing the energy demand above $(5/3)T$.

The container C_{cont} and the job j_v remain to be placed. For $k \in \{2, 3\}$, we set C_{cont} at $\sigma'(C_{\text{cont}}) = c(S'_k)$ and the job $\sigma'(j_v) = 0$, while for $k \in \{4, 5\}$, we set $\sigma'(C_{\text{cont}}) = 0$ and $\sigma'(j_v) = c(S'_k)$. We will now see, for each segment, that the peak energy demand of $(5/3)T$ is not exceeded by this new schedule.

First note that $p(j_v) \leq \gamma D \leq D/8 - \gamma D$ and hence does not intersect S'_2, when scheduled at $\sigma'(j_v) = 0$. Similarly, it is more narrow than S'_5 and $\sigma(S'_5)/2$, and hence fits right of S'_4 and S'_5 without increasing the schedule more than $(5/3)T$.

Let us now check the conditions for C_{cont}: For $k \in \{2,3\}$, we have to ensure that $c(S'_k)+p(C_{\text{cont}})/2 \leq D/2+\sigma(S'_k)$, while for $k \in \{4,5\}$ we have to prove that $p(C_{\text{cont}}) \leq \sigma(S'_k)$. It holds that $p(S'_2) \leq \frac{(15-24\gamma)D}{64} - \frac{D}{8} = \frac{(7-24\gamma)D}{64}$ and hence $p(C_{\text{cont}}) \leq 3(\frac{(7-24\gamma)D}{64})$. Therefore, $c(S'_2)+p(C_{\text{cont}}) \leq \frac{D}{2} + \frac{\sigma(S'_2)}{2}$. Furthermore, $p(S'_3) \leq \frac{(9+14\gamma)D}{32} - \frac{(15-24\gamma)D}{64} = (\frac{3}{64} + \frac{52\gamma}{32})D$ and hence $p(C_{\text{cont}}) \leq 3(\frac{3}{64} + \frac{52\gamma}{32})D$. Therefore, $c(S'_3) + p(C_{\text{cont}}) \leq \frac{(9+14\gamma)D}{32} + 3(\frac{3}{64} + \frac{52\gamma}{32})D = \frac{(27+184\gamma)D}{64}$, while $D/2 + \sigma(S'_3)/2 \geq D/2 + (\frac{(15-24\gamma)}{64} - \gamma)D/2 = (\frac{79}{128} - \frac{11}{8}\gamma)D$. As a consequence, $c(S'_3) + p(C_{\text{cont}}) \leq D/2 + \sigma(S'_3)/2$, since $\gamma \leq 1/40 \leq 25/392$.

Finally, we have $p(S'_4) \leq \frac{(3+2\gamma)D}{8} - \frac{(9+14\gamma)D}{32} = \frac{(3-6\gamma)D}{32}$. Hence, $p(C_{\text{cont}}) \leq 3\left(\frac{(3-6\gamma)D}{32}\right) = \frac{(9+14\gamma)D}{32} - \gamma D \leq \sigma(S'_4)$. While it holds that $p(S'_5) \leq \frac{D}{2} - \frac{(3+2\gamma)D}{8} = \frac{(1+2\gamma)D}{8}$. Therefore, $p(C_{\text{cont}}) \leq 3 \cdot \frac{(1+2\gamma)D}{8} = \frac{(3+6\gamma)D}{8} = \frac{(1+2\gamma)D}{8} - \gamma D \leq \sigma(S'_5)$.

For a visual representation of this repacking procedure see Figure 4. In all the cases the generated schedule has a height of at most $(5/3)T \leq (5/3)(1+\varepsilon')\mathrm{OPT} \leq (5/3+\varepsilon)\mathrm{OPT}$. ◀

4 AEPTAS for NPDM

In this section, we will prove the following theorem.

▶ **Theorem 10.** *Let $\varepsilon > 0$. There is an algorithm that places almost all jobs such that the peak energy demand is bounded by $T' := (1+\mathcal{O}(\varepsilon))\mathrm{OPT}$. For the residual jobs, we can choose one of the following containers for them to be placed in: C_1 with processing time εD and energy demand T' or a container C_2 with processing time D and energy demand $e_{\max}$. The time complexity of this algorithm is bounded by $\mathcal{O}(n\log(n)/\varepsilon) + 1/\varepsilon^{1/\varepsilon^{\mathcal{O}(1/\varepsilon)}}$.*

The statement, in fact, gives two variants of the algorithm. The first variant where all residual jobs are placed in C_1 is used in our $5/3+\varepsilon$ approximation algorithm, where the second variant with all residual jobs in C_2 can be used to obtain the AEPTAS by setting $\sigma(C_2) = 0$. The described algorithm follows the dual-approximation framework. We describe an algorithm that given a bound on the schedule peak energy demand T computes a schedule with peak energy demand $(1+\mathcal{O}(\varepsilon))T' + e_{\max}$ or decides correctly that there is no schedule with peak energy demand at most T'. This algorithm then can be called in binary search fashion with values T between $T' = \max\{T_1, T_2, T_3, T_4\}$ and $\max\{2\mathrm{work}(\mathcal{J})/D, 2e_{\max}\}$, using only multiples of $\varepsilon T'$. Note that if $e_{\max} \leq \mathcal{O}(\varepsilon^3 T')$, we can use the algorithm in [11] to find an $(1+\varepsilon)\mathrm{OPT} + \mathcal{O}(\log(1/\varepsilon)/\varepsilon \cdot \varepsilon^3 T') = (1+\mathcal{O}(\varepsilon))\mathrm{OPT}$ approximation. Hence we can assume that $e_{\max} > \mathcal{O}(\varepsilon^3 T')$.

Classification of Jobs

Given two values δ and μ with $\mu < \delta$, we partition the jobs into five sets: large, horizontal, vertical, small, and medium sized jobs. We define $\mathcal{J}_{large} := \{i \in \mathcal{J} | e(i) \geq \delta T', p(i) > \delta D\}$, $\mathcal{J}_{hor} := \{i \in \mathcal{J} | e(i) < \mu T', p(i) > \delta D\}$, $\mathcal{J}_{ver} := \{i \in \mathcal{J} | e(i) \geq \delta T', p(i) < \mu D\}$, $\mathcal{J}_{small} := \{i \in \mathcal{J} | e(i) < \mu T', p(i) < \mu D\}$, and $\mathcal{J}_{medium} := \mathcal{J} \setminus (\mathcal{J}_{large} \cup \mathcal{J}_{hor} \cup \mathcal{J}_{ver} \cup \mathcal{J}_{small})$.

▶ **Lemma 11.** *In $\mathcal{O}(n + 1/\varepsilon^2)$ operations it is possible to find values $\geq \varepsilon^{\mathcal{O}(1/\varepsilon^2)}$ for δ and μ such that $\mathrm{work}(\mathcal{J}_{medium}) \leq (\varepsilon^2/4)DT$ and $\mu \leq c\varepsilon^5\delta$ for any given constant c.*

Proof. Consider the sequence $\rho_0 := \varepsilon^5/4$, $\rho_{i+1} := c\rho_i\varepsilon^3$. Due to the pigeonhole principle, there exists an $i \in \{0, \dots, 8/\varepsilon^2\}$ such that when defining $\delta := \sigma_i$ and $\mu := \sigma_{i+1}$ the total amount of work of the medium sized jobs is bounded by $(\varepsilon^2/4)DT$, because each job appears only in two possible sets of medium jobs. We have $\delta \geq \mu \geq \varepsilon^{\mathcal{O}(1/\varepsilon^2)}$. ◀

▶ **Lemma 12.** *[43] We can round the energy demands $e(i)$ of the vertical and large jobs to multiples $k_i\varepsilon\delta T$ with $k_i \in \{1/\varepsilon, \dots, 1/\varepsilon\delta\}$ such that the number of different demands is bounded by $O(1/\varepsilon^2 \log(1/\delta))$. This rounding increases the optimal energy demand by at most $2\varepsilon T$*

Profile for vertical jobs

In the following we will dismiss the medium jobs from the schedule. Given an optimal schedule, we partition the schedule into $1/\gamma$ segments of processing time γD, for a constant $\gamma \in \mathcal{O}_\varepsilon(1)$. Given a schedule of jobs J, we define profile of J to be $\{(x, y)|y = \sum_{j\in J|\sigma(j)\leq x\leq\sigma(j)+p(j)} e(j), 0 \leq x \leq D, \}$. Energy demand of profile of jobs J at time t is $\mathcal{E}_J(t) := \sum_{j\in J|\sigma(j)\leq t\leq\sigma(j)+p(j)} e(j)$. Now consider the profile of large and horizontal jobs. Let $\tilde{J} := \mathcal{J}_{large} \cup \mathcal{J}_{hor}$, We search for the segments where the maximal energy demand of the profile of large and horizontal jobs and the minimal energy demand of this profile differs more than εT, i.e., if in segment $S := (t_a, t_b)$, $|\max_{t\in S} \mathcal{E}_{\tilde{J}}(t) - \min_{t\in S} \mathcal{E}_{\tilde{J}}(t)| \geq \varepsilon T$, then we remove all vertical and small jobs from these segments fractionally, i.e., we slice jobs, which are cut by the borders of the segment.

▷ Claim 1. Let $\mathcal{J}_{rem}$ be the set of removed vertical and small jobs. Then work$(\mathcal{J}_{rem})$ is bounded by $\mathcal{O}(\gamma/\varepsilon\delta) \cdot D \cdot T$.

Proof. Note that the energy demand of the profile of horizontal or large jobs only changes, when horizontal or large jobs end or start. The large and horizontal jobs have a total energy demand of at most T/δ since they have a processing time of at least δD and the total area of the schedule is bounded by $T \cdot D$. Hence there can be at most $2(T/\delta)/\varepsilon T = \mathcal{O}(1/\varepsilon\delta)$ segments, where the energy demand of the profile changes more than εT. As a result, the total area of the removed vertical jobs can be bounded by $\mathcal{O}(1/\varepsilon\delta) \cdot (\gamma D \cdot T)$. ◁

▷ Claim 2 (Size of γ). In the case of container C_1, we can choose $\gamma \in \mathcal{O}(\varepsilon\delta\lambda)$ such that we can schedule the removed vertical jobs fractionally inside a container $C_{1,1/4}$ of processing time $p(C_1)/4$ and energy demand $e(C_1)$. Otherwise, we can choose $\gamma \in \mathcal{O}(\varepsilon^4\delta)$ such that we can schedule the removed vertical jobs fractionally inside a container $C_{2,1/4}$ of processing time $p(C_2)/4$ and energy demand $e(C_2)$.

Proof. Let $k \in \{1, 2\}$ depending on the chosen container. First we place all the jobs $\mathcal{J}_{rem,tall}$, i.e., jobs in $\mathcal{J}_{rem}$ with energy demand larger than $e(C_{k,1/4})/2$ next to each other. The total processing time of these jobs is bounded by $2 \cdot \text{work}(\mathcal{J}_{rem,tall})/e(C_{k,1/4})$. Next, we place the residual jobs $\mathcal{J}_{rem,res} := \mathcal{J}_{rem} \setminus \mathcal{J}_{rem,tall}$, which have an energy demand of at most $e(C_{k,1/4})/2$. We take slices of processing time 1 of the jobs and place them on top of each other until the energy demand $e(C_{k,1/4})/2$ is reached. Since each job has an energy demand of at most $e(C_{k,1/4})/2$ the energy demand $e(C_{k,1/4})$ is not exceeded. The total processing time of this schedule is bounded by $2\text{work}(\mathcal{J}_{rem,res})/e(C_{k,1/4}) + 1 \leq \mathcal{O}(\gamma/(\varepsilon\delta) \cdot D \cdot T)/e(C_{k,1/4})$. Hence, for $C_{1,1/4}$ the total processing time is bounded by $\mathcal{O}(\gamma/(\varepsilon\delta) \cdot D \cdot T)/T = \mathcal{O}(\gamma/(\varepsilon\delta))D$. Hence, when choosing $\gamma \in \mathcal{O}(\lambda\varepsilon\delta)$ for a suitable constant, the total processing time of this schedule is bounded by $p(C_1)/4$. Otherwise, for container $C_{2,1/4}$ the total processing time is bounded by $\mathcal{O}((\gamma/(\varepsilon\delta) \cdot D \cdot T)/\varepsilon^3 T) = \mathcal{O}(\gamma/\varepsilon^4\delta)D$. Hence, when choosing $\gamma \in \mathcal{O}(\varepsilon^4\delta)$ for a suitable constant, the total processing time of this schedule is bounded by $p(C_2)/4$. ◁

Algorithm to place the vertical, small, and medium jobs

In the algorithm, we first round the energy demands of the vertical jobs to at most $\mathcal{O}(1/\varepsilon^2 \cdot \log(1/\delta)) = (1/\varepsilon)^{\mathcal{O}(1)}$ sizes using Lemma 12 (geometric rounding).

Afterward, we guess for each of the $1/\gamma$ segments the energy demand reserved for the vertical and small jobs rounding up to the next multiple of εT, adding at most one more εT to the energy demand of the schedule. There are at most $\mathcal{O}((1/\varepsilon)^{1/\gamma})$ possible guesses. Furthermore, we introduce one segment $\top$ of energy demand $\lceil e(C_k)/(\varepsilon T)\rceil \cdot \varepsilon T$ and processing time $p(C_k)/4$ ($k \in \{1,2\}$) for the set of removed vertical jobs. Let S_{ver} be the set of all introduced segments, and for each $s \in S_{ver}$ let $e_{s,ver}$ be the energy demand reserved for vertical and small jobs. Note that for each $s \in S_{ver}$ there exists an $i \in \{0,\dots,1/\varepsilon+3\}$ such that $e_{s,ver} = i\varepsilon T$. Furthermore, let $S_{ver,e}$ be the set of segments that have exactly energy demand e and let $p(S_{ver,e})$ be their total processing time.

To place the vertical jobs into the segments S_{ver}, we use a configuration LP. Let $C = \{a_\eta : \eta | \eta \in \{e(j)|j \in \mathcal{J}_{ver}\}\}$ be a configuration for vertical jobs, where a_η denotes the multiplicity with which the energy demand η is contained in C. We denote by $e(C) := \sum_{\eta\in\{e(j)|j\in\mathcal{J}_{ver}\}} a_\eta \cdot \eta$ the energy total demand of C, and by $\mathcal{C}_e$ the set of configurations with energy demand at most e. Furthermore, for a given configuration C we denote by $a_\eta(C)$ the number of jobs contained in C that have an energy demand of η. Since each vertical job has a energy demand of at least δT, there are at most $(1/\varepsilon)^{\mathcal{O}(1/\delta)}$ different configurations. Consider the following linear program:

$$\sum_{C\in\mathcal{C}_{i\varepsilon T}} x_{C,i} = p(S_{ver,i\varepsilon T}) \qquad \forall i \in \{1,\dots,1/\varepsilon+3\} \tag{1}$$

$$\sum_{s\in S}\sum_{C\in\mathcal{C}_{e_{s,ver}}} a_\eta(C)x_{C,s} = \sum_{j\in\mathcal{J}_{ver},e(j)=\eta} p(j) \qquad \forall \eta \in \{e(j)|j\in\mathcal{J}_{ver}\} \tag{2}$$

$$x_{C,i} \geq 0 \qquad \forall C \in \mathcal{C}, i \in \{1,\dots,1/\varepsilon\} \tag{3}$$

The variable $x_{C,i}$ represents the processing-time of configuration C inside segments $s \in S_{ver}$ with reserved energy capacity $e_{s,ver} = i\varepsilon T$. The first equation ensures that the total processing-time assigned to configurations inside segments with a certain energy capacity does not exceed the total processing time of these segments. The second equation ensures that each job is fully scheduled. More precisely, that the total processing time of jobs with a certain energy demand is covered by the configurations. A basic solution has at most $(1/\varepsilon + |\{e(j)|j \in \mathcal{J}_{ver}\}| + 1) = (1/\varepsilon)^3$ nonzero components. We can solve the above linear program by guessing the set of non zero components and then solving the resulting LP in $((1/\varepsilon)^{\mathcal{O}(1/\delta)})^{(1/\varepsilon)^3}$ time.

To place the vertical jobs, we first fill them greedily inside the configurations (slicing when the corresponding configuration slot is full) and afterwards place the configurations inside the schedule, slicing the jobs at the segment borders. For each nonzero component we have one configuration that contains at most $2/\delta$ fractionally placed vertical jobs on top of each other, which have a total energy demand of at most $2T'$. Additionally, for each segment we have the same amount of fractional jobs. Hence total area of fractionally placed jobs can be bounded by $\mu D \cdot 2T \cdot ((1/\varepsilon)^3 + 1/\gamma)$. If we choose C_1 this can be bounded by $\mathcal{O}(\mu/(\lambda\varepsilon\delta))DT \leq \lambda DT/8$, since $\mu = c\varepsilon\delta\lambda^2$ and otherwise by $\in \mathcal{O}(\mu D \cdot T/(\varepsilon^5\delta)) \leq DT/8$, since $\mu = c\varepsilon^5\delta$ for a suitable small constant c. We remove the fractionally placed jobs $\mathcal{J}_{frac}$.

Next, we place the small jobs inside the empty area that can appear above each configuration for vertical jobs. Note that there are at most $((1/\varepsilon)^{\mathcal{O}(1)} + 1/\gamma)$ configurations and the free area inside these configurations has at least the size of the total area of the small jobs. As a consequence, we have at most $2/\gamma$ rectangular areas to place the small jobs, which have a total area, which is at least the size of the small jobs. We use the NFDH algorithm to place these jobs inside the boxes until no other job fits inside.

Assume we could not place all the small jobs inside these boxes. When considering the area of free energy in each box, there are three parts that contribute to it. First, each box can have a free strip at its end, which has a processing time of at most μD. The total area of free energy contributed by this strip is bounded by $(2/\gamma)\mu D \cdot 2T$. Second, each box can have a free strip of energy demand at most μT on the top because otherwise, another line of jobs would have fitted inside this box. Since there are no boxes on top of each other, we can bound the total area of free energy inside this strip by $\mu T \cdot D$. Finally, there can be free energy between the shelves of the jobs generated by the NFDH algorithm. This total free energy is bounded by the energy demand of the tallest job times the processing time of the widest box, i.e., $\mu T \cdot \gamma D$. Hence the total area of free energy inside the boxes is bounded by $5\mu D \cdot T \cdot \gamma + D \cdot \mu T$. Since $\gamma \in \mathcal{O}(1/\varepsilon^5\delta)$ and we have chosen $\mu \leq c\delta\varepsilon^6$ for a suitable small constant $c \in \mathbb{Q}$, the total work of the remaining small jobs $\mathcal{J}_{small,res}$, which could not be placed is bounded by εTD.

We place the residual small jobs $\mathcal{J}_{small,res}$ on top of the schedule using NFDH. This adds an energy demand of at most $2\varepsilon T$ to the schedule. Next we place the medium jobs. We start all the medium jobs, that have a processing time larger than $p(C_k)/4$ with Steinberg's algorithm inside a box of energy demand at most $\mathcal{O}(\varepsilon)T$ and processing time D. This is possible since they have processing time in $(\varepsilon D, D]$ and therefore each has an energy demand of at most $\mathcal{O}(\varepsilon)T$ because their total work is bounded by $(\varepsilon^2/4)DT$. The residual jobs (that might have a processing time larger than εT) are placed inside the first half of the container using Steinberg's algorithm. The later half of the container is filled with the extra box for vertical jobs defined for the LP and the fractionally scheduled jobs. The extra box has a width of at most $p(C_k)/4$. Since the fractionally placed vertical jobs $\mathcal{J}_{frac}$ have an area of at most $e(C_k) \cdot p(C_k)/8$ and each has a width of at most $\mu D < p(C_k)/8$, we can use Steinberg's algorithm to place them inside the last quarter of the container C_k.

Placement of horizontal jobs

In this section, we first reduce the number of possible starting points for horizontal jobs and then use a linear program to place the jobs in the schedule.

First step: use geometric grouping to reduce the number of processing times of horizontal jobs. At a loss of at most $2\varepsilon T$ in the approximation ratio, we can reduce the number of processing times of horizontal jobs to $\mathcal{O}(\log(1/\delta)/\varepsilon)$ using geometric grouping (see [33, Theorem 2] by Karmarkar and Karp). These rounded jobs can be placed fractionally instead of the original jobs and an extra box of energy demand at most $\mathcal{O}(\varepsilon)T$. In this fractional packing, the horizontal jobs are sliced along the axis of the processing-time, i.e., different fractions of a job might have different starting points, but a fraction that is started, will not be interrupted and require the same amount of energy during its procession. We denote the rounded processing-time of a job j as $p'(j)$.

In the next step, we will reduce the number of starting points of the large and fractionally placed horizontal jobs without exceeding the given profile. Remember, we know the profile of large and horizontal jobs with precision εT for the segments of processing time γD.

▷ Claim 3. Without loss in the approximation ratio, we can reduce the number of different starting points of rounded horizontal and large jobs to $(1/\varepsilon)^{(1/\varepsilon)^{\mathcal{O}(1/\varepsilon)}}$.

Proof. Consider the large and horizontal jobs starting in the first segment. Since this segment has a processing time of $\gamma D \leq \delta D$, there can be no job ending in this segment. Hence this segment is maximally filled at the point γD. We can shift the start point of each job in this segment to 0 and we will not change the maximal energy demand of this segment.

Now consider a job $i \in \mathcal{J}_{hor} \cup \mathcal{J}_{large}$ starting in the second segment. If there is no horizontal or large job ending before the start of i, we can shift the start point of i to γD without changing the maximal filling energy demand in this segment. However, if there is a job $j \in \mathcal{J}_{hor} \cup \mathcal{J}_{large}$ ending before i in this segment, we can not shift this job to γD since then i and j overlap, which they did not before. This could change the maximal energy demand of the profile in this segment. Nevertheless, if j is the last job ending before i, we can shift i to the left, such that i starts at the endpoint of j.

We iterate this shifting with all segments and all jobs in $\mathcal{J}_{hor} \cup \mathcal{J}_{large}$. As a result, all jobs start either at a multiple of γD, or they start at an endpoint of an other job in $\mathcal{J}_{hor} \cup \mathcal{J}_{large}$. Therefore, we can describe the set of possible starting points for jobs in $\mathcal{J}_{hor} \cup \mathcal{J}_{large}$ as $S_{hor,large} := \{l\gamma D + \sum_{j=1}^{1/\delta} p(i_j) | l \in \{0, 1, \ldots, 1/\gamma\}, i_j \in \mathcal{J}_{hor} \cup \mathcal{J}_{large} \forall j \in \{1, \ldots, 1/\delta_w\}\}$. It holds that $|S_{hor,large}| \leq (1/\gamma) \cdot (\log(1/\delta)/\epsilon)^{1/\delta} = (1/\varepsilon)^{(1/\varepsilon)^{\mathcal{O}(1/\varepsilon)}}$. ◁

▷ Claim 4. At a loss of at most $\mathcal{O}(\varepsilon T)$ in the approximation ratio, we can reduce the number of *used* starting points for rounded horizontal jobs to $\mathcal{O}(1/\varepsilon\delta)$.

Proof. We partition the set of horizontal jobs by their processing time into $\mathcal{O}(\log(1/\delta))$ sets $\mathcal{J}_{hor}^l := \{i \in \mathcal{J}_{hor} | D/2^l < p(i) \leq D/2^{l-1}\}$. For each of these sets, we will reduce the number of starting positions to $2^l/\varepsilon^2$. We partition the schedule into 2^l segments of processing time $D/2^l$. Each job from the set $\mathcal{J}_{hor}^l$ has a processing time larger than $D/2^l$ and hence it starts in an other segment as it ends. We consider for each segment all the horizontal jobs of the set $\mathcal{J}_{hor}^l$ ending in this segment and sort them by increasing starting position. Let $e_{l,i}$ be the energy demand of the stack of jobs in $\mathcal{J}_{hor}^l$ ending in the i-th segment. We partition the stack into $1/\varepsilon$ layers of energy demand $\varepsilon e_{l,i}$ and slice the horizontal jobs overlapping the layer borders. We remove all the jobs in the bottom most layer and shift the jobs from the layers above to the left, such that they start at the latest original start position from the layer below. We repeat this procedure for each segment. By this shift, we reduce the total number of starting positions from jobs from the set $\mathcal{J}_{hor}^l$ to $2^l/\varepsilon$. The total energy demand of the jobs we removed is bounded by $\varepsilon e(\mathcal{J}_{hor}^l)$. Since these jobs have a processing time of at most $D/2^{l-1}$, we can schedule 2^{l-1} of these jobs after one an other (horizontally), without violating the deadline. Hence, when scheduling these jobs fractionally, we add at most $\varepsilon e(\mathcal{J}_{hor}^l)/2^{l-1}$ to the schedule. Note that since all the jobs in set $\mathcal{J}_{hor}^l$ have a processing time of at least $D/2^l$, it holds that $\sum_{l=1}^{\lceil \log(1/\delta) \rceil} e(\mathcal{J}_{hor}^l)/2^l \leq T$ and, hence, we add at most $\sum_{l=1}^{\lceil \log(1/\delta) \rceil} \varepsilon e(\mathcal{J}_{hor}^l)/2^{l-1} \leq 2\varepsilon T$ to the energy demand of the schedule, when scheduling the removed horizontal jobs. The total number of starting positions is bounded by $\sum_{l=1}^{\lceil \log(1/\delta) \rceil} 2^l/\varepsilon = (2^{\lceil \log(1/\delta) \rceil + 1} - 1)/\varepsilon \in \mathcal{O}(1/\delta_w \varepsilon)$ ◁

Algorithm to place horizontal and large jobs

To place the jobs in $\mathcal{J}_{hor} \cup \mathcal{J}_{large}$, we first guess the starting positions of the large jobs $\mathcal{J}_{large}$ in $\mathcal{O}(|S_{hor,large}|^{|\mathcal{J}_{large}|}) = (1/\varepsilon)^{(1/\varepsilon)^{\mathcal{O}(1/\varepsilon)}}$. Note that this guess affects the energy demand that is left for horizontal jobs. Next we guess which $\mathcal{O}(1/\varepsilon\delta)$ starting points in $S_{hor,large}$ will be used after the shifting due to Claim 4. There are at most $|S_{hor,large}|^{\mathcal{O}(1/\varepsilon\delta)} = (1/\varepsilon)^{(1/\varepsilon)^{\mathcal{O}(1/\varepsilon)}}$ possible guesses total. We call the set of guessed starting points $\bar{S}_{h,l}$. For each starting point in $\bar{S}_{h,l}$, we calculate the residual total energy demand, that is left after the guess for the large jobs. For a given $s \in \bar{S}_{h,l}$ let $e_{s,hor}$ be this residual total energy demand.

Consider the following linear program for horizontal jobs:

$$\sum_{\rho\in\{p'(j)|j\in\mathcal{J}_{hor}\}} \sum_{\substack{s'\in\bar{S}_{h,l}\\ s'\leq s<s'+\rho}} x_{\rho,s'} \leq e_{s,hor} \qquad \forall s\in\bar{S}_{h,l}$$

$$\sum_{s\in\bar{S}_{h,l}} x_{\rho,s} = \sum_{j\in\mathcal{J}_{hor},p'(j)=\rho} e(j) \qquad \forall \rho\in\{p'(j)|j\in\mathcal{J}_{hor}\}$$

$$x_{\rho,s}\geq 0 \qquad \forall s\in\bar{S}_{h,l},\rho\in\{p'(j)|j\in\mathcal{J}_{hor}\}$$

The variable $x_{\rho,s}$ denotes the total energy demand of jobs with rounded processing time ρ starting at s. The first equation ensures that the energy capacity at a starting time s is not exceeded by the jobs starting at or overlapping s. The second equation ensures that the total energy requirement of jobs with rounded processing time ρ is covered by energy demand of jobs with this processing time started in the schedule.

A basic solution to this linear program has at most $|\bar{S}_{h,l}|+|\mathcal{J}_{hor}|=\mathcal{O}(1/\varepsilon\delta)$ non zero components. We can guess the non zero components in at most $(|\bar{S}_{h,l}|\cdot|\bar{\mathcal{J}}_{hor}|)^{|\bar{S}_{h,l}|+|\bar{\mathcal{J}}_{hor}|}=(1/\varepsilon)^{(1/\varepsilon)^{\mathcal{O}(1/\varepsilon)}}$. Furthermore, we can guess their value with precision μT in at most $(1/\mu)^{|\bar{S}_{h,l}|+|\bar{\mathcal{J}}_{hor}|}=(1/\varepsilon)^{(1/\varepsilon)^{\mathcal{O}(1/\varepsilon)}}$ guesses. Scheduling all the horizontal jobs integral and the error due to the precision add at most $2\mu T\cdot(|\bar{S}_{h,l}|+|\bar{\mathcal{J}}_{hor}|)$ to the peak energy demand. Note that $2\mu T\cdot(|\bar{S}_{h,l}|+|\bar{\mathcal{J}}_{hor}|)\leq\mathcal{O}(\varepsilon)T'$ since $\mu\leq\mathcal{O}(\varepsilon^2\delta)$.

After this step, we either have scheduled all given jobs or have decided that it is not possible for the given guess of T and the profile. If it is not possible for any profile, we have to increase T. If we have found a schedule, we try the next smaller value for T. Each of the steps has increased the peak energy demand by at most $\mathcal{O}(\varepsilon)T$ above T. Besides of the job classification and rounding, each step of the algorithm is bounced by $(1/\varepsilon)^{1/\varepsilon^{\mathcal{O}(1/\varepsilon)}}$. Therefore, the described algorithms fulfills the claims of Theorem 10.

5 Conclusion

In this paper, we presented an AEPTAS with additive term $e_{\max}$ as well as a $(5/3+\varepsilon)$-approximation for Nonpreemptive Peak Demand Minimization (NPDM). Since the lower bound for approximation algorithms for this problem is known to be 3/2, this leaves a small gap between the lower bound and the approximation guarantee. Closing this gap is an interesting open question for further research, especially since for the related strip packing problem the same gap is yet to be resolved.

References

1. Anna Adamaszek, Sariel Har-Peled, and Andreas Wiese. Approximation schemes for independent set and sparse subsets of polygons. *J. ACM*, 66(4):29:1–29:40, 2019.
2. Anna Adamaszek, Tomasz Kociumaka, Marcin Pilipczuk, and Michal Pilipczuk. Hardness of approximation for strip packing. *ACM Transactions on Computation Theory*, 9(3):14:1–14:7, 2017. doi:10.1145/3092026.
3. Soroush Alamdari, Therese Biedl, Timothy M Chan, Elyot Grant, Krishnam Raju Jampani, Srinivasan Keshav, Anna Lubiw, and Vinayak Pathak. Smart-grid electricity allocation via strip packing with slicing. In *Workshop on Algorithms and Data Structures*, pages 25–36. Springer, 2013.
4. Brenda S. Baker, Edward G. Coffman Jr., and Ronald L. Rivest. Orthogonal packings in two dimensions. *SIAM J. Comput.*, 9(4):846–855, 1980. doi:10.1137/0209064.

5 Nikhil Bansal, Alberto Caprara, and Maxim Sviridenko. A new approximation method for set covering problems, with applications to multidimensional bin packing. *SIAM J. Comput.*, 39(4):1256–1278, 2009. doi:10.1137/080736831.

6 Nikhil Bansal, José R. Correa, Claire Kenyon, and Maxim Sviridenko. Bin packing in multiple dimensions: Inapproximability results and approximation schemes. *Math. Oper. Res.*, 31(1):31–49, 2006. doi:10.1287/moor.1050.0168.

7 Nikhil Bansal and Arindam Khan. Improved approximation algorithm for two-dimensional bin packing. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 13–25. SIAM, 2014. doi:10.1137/1.9781611973402.2.

8 Nikhil Bansal, Andrea Lodi, and Maxim Sviridenko. A tale of two dimensional bin packing. In *FOCS*, pages 657–666, 2005.

9 Iwo Błądek, Maciej Drozdowski, Frédéric Guinand, and Xavier Schepler. On contiguous and non-contiguous parallel task scheduling. *Journal of Scheduling*, 18(5):487–495, 2015.

10 Marin Bougeret, Pierre François Dutot, Klaus Jansen, Christina Otte, and Denis Trystram. Approximating the non-contiguous multiple organization packing problem. In *IFIP International Conference on Theoretical Computer Science*, pages 316–327. Springer, 2010.

11 Marin Bougeret, Pierre-François Dutot, Klaus Jansen, Christina Robenek, and Denis Trystram. Approximation algorithms for multiple strip packing and scheduling parallel jobs in platforms. *Discret. Math. Algorithms Appl.*, 3(4):553–586, 2011. doi:10.1142/S1793830911001413.

12 Nilotpal Chakraborty, Arijit Mondal, and Samrat Mondal. Efficient scheduling of nonpreemptive appliances for peak load optimization in smart grid. *IEEE Transactions on Industrial Informatics*, 14(8):3447–3458, 2017.

13 Henrik I. Christensen, Arindam Khan, Sebastian Pokutta, and Prasad Tetali. Approximation and online algorithms for multidimensional bin packing: A survey. *Computer Science Review*, 24:63–79, 2017.

14 Waldo Gálvez, Fabrizio Grandoni, Afrouz Jabal Ameli, Klaus Jansen, Arindam Khan, and Malin Rau. A tight $(3/2+\epsilon)$ approximation for skewed strip packing. In Jaroslaw Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, volume 176 of *LIPIcs*, pages 44:1–44:18, 2020.

15 Waldo Gálvez, Fabrizio Grandoni, Afrouz Jabal Ameli, and Kamyar Khodamoradi. Approximation algorithms for demand strip packing. *CoRR*, abs/2105.08577, 2021. arXiv:2105.08577.

16 Waldo Gálvez, Fabrizio Grandoni, Sandy Heydrich, Salvatore Ingala, Arindam Khan, and Andreas Wiese. Approximating geometric knapsack via l-packings. In *FOCS*, pages 260–271, 2017.

17 Waldo Gálvez, Fabrizio Grandoni, Arindam Khan, Diego Ramirez-Romero, and Andreas Wiese. Improved approximation algorithms for 2-dimensional knapsack: Packing into multiple l-shapes, spirals and more. In *SoCG*, pages 39:1–39:17, 2021.

18 Waldo Gálvez, Fabrizio Grandoni, Salvatore Ingala, and Arindam Khan. Improved pseudo-polynomial-time approximation for strip packing. In *36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 65, pages 9:1–9:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.FSTTCS.2016.9.

19 Fabrizio Grandoni, Tobias Mömke, Andreas Wiese, and Hang Zhou. A $(5/3+\varepsilon)$-approximation for unsplittable flow on a path: placing small tasks into boxes. In *STOC*, pages 607–619, 2018.

20 Rolf Harren, Klaus Jansen, Lars Prädel, and Rob van Stee. A (5/3 + eps)-approximation for 2d strip packing. In Andreas Brieden, Zafer-Korcan Görgülü, Tino Krug, Erik Kropat, Silja Meyer-Nieberg, Goran Mihelcic, and Stefan Wolfgang Pickl, editors, *11th Cologne-Twente Workshop on Graphs and Combinatorial Optimization, Munich, Germany, May 29-31, 2012. Extended Abstracts*, pages 139–142, 2012.

21 Rolf Harren and Rob van Stee. Improved absolute approximation ratios for two-dimensional packing problems. In Irit Dinur, Klaus Jansen, Joseph Naor, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 12th International Workshop, APPROX 2009, and 13th International Workshop, RANDOM 2009, Berkeley, CA, USA, August 21-23, 2009. Proceedings*, volume 5687 of *Lecture Notes in Computer Science*, pages 177–189. Springer, 2009. doi:10.1007/978-3-642-03685-9_14.

22 Sören Henning, Klaus Jansen, Malin Rau, and Lars Schmarje. Complexity and inapproximability results for parallel task scheduling and strip packing. *Theory of Computing Systems*, 64(1):120–140, 2020. doi:10.1007/s00224-019-09910-6.

23 Klaus Jansen. Scheduling malleable parallel tasks: An asymptotic fully polynomial time approximation scheme. *Algorithmica*, 39(1):59–81, 2004.

24 Klaus Jansen. A $(3/2+\varepsilon)$ approximation algorithm for scheduling moldable and non-moldable parallel tasks. In *Proceedings of the twenty-fourth annual ACM symposium on Parallelism in algorithms and architectures*, pages 224–235, 2012.

25 Klaus Jansen and Felix Land. Scheduling monotone moldable jobs in linear time. In *2018 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2018, Vancouver, BC, Canada, May 21-25, 2018*, pages 172–181. IEEE Computer Society, 2018. doi:10.1109/IPDPS.2018.00027.

26 Klaus Jansen and Lars Prädel. A new asymptotic approximation algorithm for 3-dimensional strip packing. In *40th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, volume 8327, pages 327–338. Springer, 2014. doi:10.1007/978-3-319-04298-5_29.

27 Klaus Jansen and Malin Rau. Closing the gap for pseudo-polynomial strip packing. In *ESA*, volume 144, pages 62:1–62:14, 2019.

28 Klaus Jansen and Roberto Solis-Oba. Rectangle packing with one-dimensional resource augmentation. *Discret. Optim.*, 6(3):310–323, 2009. doi:10.1016/j.disopt.2009.04.001.

29 Klaus Jansen and Ralf Thöle. Approximation algorithms for scheduling parallel jobs. *SIAM J. Comput.*, 39(8):3571–3615, 2010. doi:10.1137/080736491.

30 Klaus Jansen and Guochuan Zhang. Maximizing the total profit of rectangles packed into a rectangle. *Algorithmica*, 47(3):323–342, 2007. doi:10.1007/s00453-006-0194-5.

31 Edward G. Coffman Jr., M. R. Garey, David S. Johnson, and Robert Endre Tarjan. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM J. Comput.*, 9(4):808–826, 1980. doi:10.1137/0209062.

32 Mohammad M Karbasioun, Gennady Shaikhet, Ioannis Lambadaris, and Evangelos Kranakis. Asymptotically optimal scheduling of random malleable demands in smart grid. *Discrete Mathematics, Algorithms and Applications*, 10(02):1850025, 2018.

33 Narendra Karmarkar and Richard M. Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 312–320. IEEE Computer Society, 1982.

34 Claire Kenyon and Eric Rémila. A near-optimal solution to a two-dimensional cutting stock problem. *Math. Oper. Res.*, 25(4):645–656, 2000. doi:10.1287/moor.25.4.645.12118.

35 Arindam Khan, Arnab Maiti, Amatya Sharma, and Andreas Wiese. On guillotine separable packings for the two-dimensional geometric knapsack problem. In *SoCG*, pages 48:1–48:17, 2021.

36 Arindam Khan and Madhusudhan Reddy Pittu. On guillotine separability of squares and rectangles. In *APPROX*, pages 47:1–47:22, 2020.

37 Fu-Hong Liu, Hsiang-Hsuan Liu, and Prudence WH Wong. Optimal nonpreemptive scheduling in a smart grid model. In *27th International Symposium on Algorithms and Computation (ISAAC 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

38 Tobias Mömke and Andreas Wiese. Breaking the barrier of 2 for the storage allocation problem. In *ICALP*, pages 86:1–86:19, 2020.

39 Giorgi Nadiradze and Andreas Wiese. On approximating strip packing with a better ratio than 3/2. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1491–1510. SIAM, 2016. doi:10.1137/1.9781611974331.ch102.

40 Anshu Ranjan, Pramod Khargonekar, and Sartaj Sahni. Offline preemptive scheduling of power demands to minimize peak power in smart grids. In *2014 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6. IEEE, 2014.

41 Anshu Ranjan, Pramod Khargonekar, and Sartaj Sahni. Offline first fit scheduling in smart grids. In *2015 IEEE Symposium on Computers and Communication (ISCC)*, pages 758–763. IEEE, 2015.

42 Anshu Ranjan, Pramod Khargonekar, and Sartaj Sahni. Smart grid power scheduling via bottom left decreasing height packing. In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 1128–1133. IEEE, 2016.

43 Malin Rau. *Useful Structures and How to Find Them: Hardness and Approximation Results for Various Variants of the Parallel Task Scheduling Problem.* dissertation, Kiel University, Kiel, Germany, 2019.

44 Ingo Schiermeyer. Reverse-fit: A 2-optimal algorithm for packing rectangles. In Jan van Leeuwen, editor, *Algorithms - ESA '94, Second Annual European Symposium, Utrecht, The Netherlands, September 26-28, 1994, Proceedings*, volume 855 of *Lecture Notes in Computer Science*, pages 290–299. Springer, 1994. doi:10.1007/BFb0049416.

45 Pierluigi Siano. Demand response and smart grids—a survey. *Renewable and sustainable energy reviews*, 30:461–478, 2014.

46 Daniel Dominic Sleator. A 2.5 times optimal algorithm for packing in two dimensions. *Inf. Process. Lett.*, 10(1):37–40, 1980. doi:10.1016/0020-0190(80)90121-0.

47 A. Steinberg. A strip-packing algorithm with absolute performance bound 2. *SIAM J. Comput.*, 26(2):401–409, 1997. doi:10.1137/S0097539793255801.

48 Shaojie Tang, Qiuyuan Huang, Xiang-Yang Li, and Dapeng Wu. Smoothing the energy consumption: Peak demand reduction in smart grid. In *32nd IEEE International Conference on Computer Communications (INFOCOM)*, pages 1133–1141. IEEE, 2013. doi:10.1109/INFCOM.2013.6566904.

49 Sean Yaw, Brendan Mumey, Erin McDonald, and Jennifer Lemke. Peak demand scheduling in the smart grid. In *2014 IEEE international conference on smart grid communications (SmartGridComm)*, pages 770–775. IEEE, 2014.

A Proof of Theorem 4

Proof: $T_2 \leq \mathrm{OPT}(I)$.

▶ **Lemma 13.** *It holds that* $p(\mathcal{J}_{e(i)>(1/3)\mathrm{OPT}}) + p(\mathcal{J}_{e(i)>(2/3)\mathrm{OPT}}) \leq 2D$ *and* $p(\mathcal{J}_{e(i)>(1/2)\mathrm{OPT}}) \leq D$.

Proof. Note that jobs with energy demand larger than (1/3)OPT cannot intersect the same vertical line as jobs with energy demand larger than (2/3)OPT in an optimal schedule. Furthermore, each vertical line through an optimal schedule can intersect at most two jobs with energy demand larger than (1/3)OPT. Moreover, no vertical line can intersect two jobs from the set $\mathcal{J}_{e(i)>(1/2)\mathrm{OPT}}$. The claim is a consequence. ◀

▶ **Corollary 14.** *The smallest value T such that* $p(\mathcal{J}_{e(i)>(1/3)T}) + p(\mathcal{J}_{e(i)>(2/3)T}) \leq 2D$ *and* $p(\mathcal{J}_{e(i)>T/2}) \leq D$ *is a lower bound for* OPT.

Proof. By Lemma 13, we know that $p(\mathcal{J}_{e(i)>(1/3)\mathrm{OPT}}) + p(\mathcal{J}_{e(i)>(2/3)\mathrm{OPT}}) \leq 2D$ and obviously we have $p(\mathcal{J}_{e(i)>\mathrm{OPT}/2}) \leq D$. Therefore, the smallest value such that $p(\mathcal{J}_{e(i)>(1/3)T}) + p(\mathcal{J}_{e(i)>(2/3)T}) \leq 2D$ and $p(\mathcal{J}_{e(i)>T/2}) \leq D$ has to be a lower bound on OPT. ◀

Note that we can find this smallest value in $\mathcal{O}(n \log n)$ by starting with $T = T_1$ and as long as $p(\mathcal{J}_{e(i)\geq 1/3T}) + p(\mathcal{J}_{e(i)\geq 2/3T}) > 2D$ or $p(\mathcal{J}_{e(i)>T/2}) > D$ update T as follows: For $l \in [0,1]$, denote by e_l the energy demand of the smallest job in $\mathcal{J}_{e(i)>l\cdot T}$ and set $T := \min\{3e_{1/3}, (3/2)e_{2/3}, 2e_{1/2}\}$. This iteratively excludes one job from one of the three sets. We denote this lower bound as

$$T_2 := \min\{T | p(\mathcal{J}_{e(i)\geq T/3}) + p(\mathcal{J}_{e(i)\geq 2T/3}) \leq 2D \wedge p(\mathcal{J}_{e(i)\geq T/2}) \leq D\}.$$

▶ **Lemma 15.** *Let $w' \in [0, 1/2)$. Then*

$$p(\mathcal{J}_{e(i)>(2/3)T_2}) > (1-w')D \Rightarrow p(\mathcal{J}_{e(i)\in((1/3)T_2,(2/3)T_2]}) \leq 2w'D.$$

Proof. We know that $p(\mathcal{J}_{e(i)>(1/3)T_2}) + p(\mathcal{J}_{e(i)>(2/3)T_2}) \leq 2D$. Because $\mathcal{J}_{e(i)>(2/3)T_2} \subseteq \mathcal{J}_{e(i)>(1/3)T_2}$ and $p(\mathcal{J}_{e(i)>(2/3)T_2}) > (1-w')D$, it holds that $p(\mathcal{J}_{e(i)\in((1/3)T_2,(2/3)T_2]}) \leq 2w'D$. ◀

Proof: $T_3 \leq \mathrm{OPT}(I)$. Next, we obtain a bound based on a set of jobs that do not overlap vertically in a given optimal schedule.

▶ **Lemma 16.** *Consider an optimal schedule and let $\mathcal{J}_{\text{seq}}$ be a set of jobs such that no pair of jobs $i, i' \in \mathcal{J}_{\text{seq}}$ overlaps vertically, i.e., $\sigma(i) + p(i) \leq \sigma(i')$ or $\sigma(i') + p(i') \leq \sigma(i)$. Furthermore, define $\mathcal{J}_w := \mathcal{J}_{p(i)>(D-p(\mathcal{J}_{\text{seq}})/2)} \setminus \mathcal{J}_{\text{seq}}$. Then there exists a vertical line through the schedule that intersects a job in $\mathcal{J}_{\text{seq}}$ and all the jobs in $\mathcal{J}_w$.*

Proof. First note that $(D - p(\mathcal{J}_{\text{seq}})/2) \geq D/2$. Consider the vertical strip between $p(\mathcal{J}_{\text{seq}})/2$ and $(D - p(\mathcal{J}_{\text{seq}}))/2$. Each job in $\mathcal{J}_w$ completely overlaps this strip. Furthermore, either the strip itself contains a job in $\mathcal{J}_{\text{seq}}$, in which case the claim is trivially true, or on each position on both sides of the strip there is a job from $\mathcal{J}_{\text{seq}}$. Assume the latter case. Since the jobs in $\mathcal{J}_w$ have a time demand strictly larger than $(D - p(\mathcal{J}_{\text{seq}})/2)$, there exists an $\sigma > 0$ such that the vertical line at $(D - p(\mathcal{J}_{\text{seq}})/2) + \sigma$ as well is overlapped by all the jobs in this set. Since this line intersects also a job from the set $\mathcal{J}_{\text{seq}}$, the claim follows. ◀

▶ **Corollary 17.** *Let $\mathcal{J}_{\text{seq}}$ be a set of jobs such that $p(\mathcal{J}_{\text{seq}}) \leq D$ and consider $\mathcal{J}_w := \mathcal{J}_{p(i)>D-p(\mathcal{J}_{\text{seq}})/2} \setminus \mathcal{J}_{\text{seq}}$. Furthermore let $i_\perp \in \mathcal{J}_{\text{seq}}$ be the job with the smallest energy demand. Then it holds that $\min\{e(i_\perp) + e(\mathcal{J}_w), 2e(i_\perp)\} \leq \mathrm{OPT}$.*

Proof. Consider an optimal solution. If two jobs from the set $\mathcal{J}_{\text{seq}}$ intersect the same vertical line, $2e(i_\perp)$ is obviously a lower bound on OPT. On the other hand, if in any optimal schedule there does not exist a pair of jobs from $\mathcal{J}_{\text{seq}}$ that overlap the same vertical line, we know by Lemma 16 that there exists a job in $\mathcal{J}_{\text{seq}}$ that overlaps with all the jobs in $\mathcal{J}_w$ and therefore $\mathrm{OPT} \geq e(i_\perp) + e(\mathcal{J}_w)$ in this case. ◀

From Corollary 17, we derive a lower bound on OPT. For a given $k \in [n]$, we define $\mathcal{J}_k$ to be the set of the k jobs with the largest energy demand in $\mathcal{J}$ and $\mathcal{J}'_k$ to be the set of the k jobs with the largest energy demand in $\mathcal{J} \setminus \mathcal{J}_{p(i)>D/2}$. Let i_k and i'_k be the jobs with the smallest energy demand in $\mathcal{J}_k$ and $\mathcal{J}'_k$, respectively. We define:

$$T_{3,a} := \max\{\min\{e(i_k) + e(\mathcal{J}_{p(i)>D-p(\mathcal{J}_k)/2} \setminus \mathcal{J}_k), 2e(i_k)\} | k \in \{1,\ldots,n\}, p(\mathcal{J}_k) \leq D\},$$
$$T_{3,b} := \max\{\min\{e(i'_k) + e(\mathcal{J}_{p(i)>D-p(\mathcal{J}'_k)/2}), 2e(i'_k)\} | k \in \{1,\ldots,n\}, p(\mathcal{J}'_k) \leq D\},$$

and finally $T_3 = \max\{T_{3,a}, T_{3,b}\}$. Note that $\mathcal{J}'_k$ and $\mathcal{J}_{p(i)>D-p(\mathcal{J}'_k)/2}$ are disjoint, since $\mathcal{J}'_k$ contains only jobs with processing time at most $D/2$ and $\mathcal{J}_{p(i)>D-p(\mathcal{J}'_k)/2}$ contains only jobs with processing time larger than $D/2$, and hence, by Corollary 17, T_3 is a lower bound on OPT. For this lower bound, we prove the following property.

▶ **Lemma 18.** *Let* $T = \max\{T_1, T_2, T_3\}$*,* $w \in (0, 1/2)$ *and* $h \in (1/2, 1]$ *as well as* $\mathcal{J}_h := \mathcal{J}_{e(i) \geq hT}$ *and* $\mathcal{J}_w := \mathcal{J}_{p(i) > (1/2 + w/2)D} \setminus \mathcal{J}_h$*. It holds that*

$$p(\mathcal{J}_h) \geq (1 - w)D \Rightarrow e(\mathcal{J}_w) \leq (1 - h)T.$$

Proof. Since $T \geq T_2$, it holds that $p(\mathcal{J}_h) \leq D$. By construction of T_3 for each job $j \in \mathcal{J}_h$, it holds that $e(j) + e(\mathcal{J}_{p(i) > D - p(\mathcal{J}_{e(i) \geq e(j)})/2} \setminus \mathcal{J}_{e(i) \geq e(j)}) \leq T_3$, because $2e(j) > T_3$ (and $T_3 \geq \min\{2e(j), e(j) + e(\mathcal{J}_{p(i) > D - p(\mathcal{J}_{e(i) \geq e(j)})/2} \setminus \mathcal{J}_{e(i) \geq e(j)})\}$). Furthermore, note that $\mathcal{J}_h = \mathcal{J}_{e(i) \geq e(j)}$ for the job j with the smallest energy demand in $\mathcal{J}_h$.

Therefore, if $p(\mathcal{J}_h) \geq (1 - w)D$, it holds that $\mathcal{J}_{p(i) > D - (1-w)D/2} \subseteq \mathcal{J}_{p(i) > D - p(\mathcal{J}_h)/2}$ and hence,

$$\begin{aligned} hT + e(\mathcal{J}_w) &= hT + e(\mathcal{J}_{p(i) > D - (1-w)D/2} \setminus \mathcal{J}_h) \\ &\leq e(j) + e(\mathcal{J}_{p(i) > D - p(\mathcal{J}_h)/2} \setminus \mathcal{J}_h) \leq T_3 \leq T. \end{aligned}$$ ◀

▶ **Lemma 19.** *Let* $T = \max\{T_1, T_2, T_3\}$*,* $w \in (1/2, 1]$ *and* $h \in (1/2, 1]$ *as well as* $\mathcal{J}_w := \mathcal{J}_{p(i) \geq wD}$ *and* $\mathcal{J}_h := \mathcal{J}_{e(i) > hT} \setminus \mathcal{J}_{p(i) > D/2}$*. It holds that*

$$e(\mathcal{J}_w) > (1 - h)T \Rightarrow p(\mathcal{J}_h) \leq 2(1 - w)D.$$

Proof. Let $e(\mathcal{J}_w) > (1 - h)T$. Since for each job in $\mathcal{J}_h$ it holds that $e(i) > T/2 \geq T_{3,b}/2$, by definition of $T_{3,b}$, for each $j \in \mathcal{J}_h$ it holds that $e(j) + e(\mathcal{J}_{p(i) > D - p(\mathcal{J}_{e(i) \geq e(j)})/2}) \leq T$. Therefore for the smallest job $j \in \mathcal{J}_h$, it holds that $e(j) + e(\mathcal{J}_{p(i) > D - p(\mathcal{J}_h)/2}) \leq T$.

For contradiction assume that $p(\mathcal{J}_h) > 2(1 - w)D$. Note that in this case $D - p(\mathcal{J}_h)/2 < wD$ and hence $e(\mathcal{J}_{p(i) > D - p(\mathcal{J}_h)/2}) \geq e(\mathcal{J}_D) > (1 - h)T$. As a consequence $e(j) + e(\mathcal{J}_{p(i) > D - p(\mathcal{J}_h)/2}) > hT + (1 - h)T = T$, a contradiction. ◀

Proof: $T_4 \leq \mathrm{OPT}(I)$.

▶ **Lemma 20.** *Consider an optimal schedule and let* $\mathcal{J}_{\mathrm{seq}} \subseteq \mathcal{J}$ *be a set of jobs such that no pair of jobs* $j, j' \in \mathcal{J}_{\mathrm{seq}}$ *overlaps vertically, i.e.,* $\sigma(j) + p(j) \leq \sigma(j')$ *or* $\sigma(j') + p(j') \leq \sigma(j)$*. Let* $\mathcal{J}_D \subseteq \mathcal{J}_{p(j) > (\max\{D - p(\mathcal{J}_{\mathrm{seq}}), D/2\})} \setminus \mathcal{J}_{\mathrm{seq}}$*. Then there exists a vertical line through the schedule that intersects a job in* $\mathcal{J}_{\mathrm{seq}}$ *and a subset* $\mathcal{J}_{W'} \subseteq \mathcal{J}_D$ *with* $e(\mathcal{J}'_D) \geq e(\mathcal{J}_D)/2$*.*

Proof. First, we consider the trivial cases. If a job from $\mathcal{J}_{\mathrm{seq}}$ overlaps the vertical line at $D/2$ the claim is trivially true, since all the jobs from $\mathcal{J}_D$ overlap $D/2$. On the other hand, if all the jobs in $\mathcal{J}_{\mathrm{seq}}$ are left or right of $D/2$, it holds that $p(\mathcal{J}_{\mathrm{seq}}) \leq D/2$ and one of the jobs has a distance of at most $D/2 - p(\mathcal{J}_{\mathrm{seq}})$ from $D/2$. This job has to be overlapped by all the jobs from $\mathcal{J}_D$ since they have a width larger than $D - p(\mathcal{J}_{\mathrm{seq}})$.

Otherwise, consider the vertical line L_l through the right border of the rightmost job from $\mathcal{J}_{\mathrm{seq}}$ that is left of $D/2$ and the vertical line L_r through the left border of the leftmost job from $\mathcal{J}_{\mathrm{seq}}$ that is right of $D/2$. Note that L_l and L_r have a distance of at most $(D - p(\mathcal{J}_{\mathrm{seq}}))$. Consider the set $\mathcal{J}_{D,l} \subseteq \mathcal{J}_D$ that is intersected by the vertical line L_l. Note that the residual jobs in $\mathcal{J}_{D,r} := \mathcal{J}_D \setminus \mathcal{J}_{D,l}$ all overlap the vertical line at $L_l + (D - p(\mathcal{J}_{\mathrm{seq}})) \geq L_r$ and hence L_r as well. Since $\mathcal{J}_{D,r} \cup \mathcal{J}_{D,l} = \mathcal{J}_D$, one of the two sets has an energy demand of at least $e(\mathcal{J}_D)/2$. Finally, note that there exists a small enough $\sigma > 0$ such that $L_l - \sigma$ and $L_r + \sigma$ overlap the same set of wide jobs as L_l and L_r as well as the corresponding job in $\mathcal{J}_{\mathrm{seq}}$. ◀

▶ **Corollary 21.** *Let* $\mathcal{J}_{\mathrm{seq}}$ *be a set of jobs such that* $p(\mathcal{J}_{\mathrm{seq}}) \leq D$ *and consider* $\mathcal{J}_D := \mathcal{J}_{p(i) > (\max\{D - p(\mathcal{J}_{\mathrm{seq}}), D/2\})} \setminus \mathcal{J}_{\mathrm{seq}}$*. Furthermore let* $i_\perp \in \mathcal{J}_{\mathrm{seq}}$ *be the job with the smallest energy demand. Then it holds that* $\min\{e(i_\perp) + e(\mathcal{J}_D)/2, 2e(i_\perp)\} \leq \mathrm{OPT}$*.*

Proof. Consider an optimal solution. If two jobs from the set $\mathcal{J}_{\text{seq}}$ intersect the same vertical line, $2e(i_\perp)$ is obviously a lower bound on OPT. Otherwise, if in any optimal schedule there does not exist a pair of jobs from $\mathcal{J}_{\text{seq}}$ that overlap the same vertical line, we know by Lemma 20 that there exists a job in $\mathcal{J}_{\text{seq}}$ that overlaps with a set $\mathcal{J}'_D \subseteq \mathcal{J}_D$ such that $e(\mathcal{J}'_D) \geq e(\mathcal{J}_D)/2$ and therefore OPT $\geq e(i_\perp) + e(\mathcal{J}_D)/2$ in this case. ◀

Define $\mathcal{J}_k$ as the set of the k jobs with largest energy demand. Furthermore, define $\mathcal{J}_{D,k} := \mathcal{J}_{p(i)>(\max\{D-p(\mathcal{J}_k),D/2\})} \setminus \mathcal{J}_k$. Let i_k be the job with the smallest energy demand in $\mathcal{J}_k$. We define the value T_4, which by Corollary 21 is a lower bound for OPT as follows:

$$T_4 := \max\{\min\{2e(i_k), e(i_k) + e(\mathcal{J}_{D,k})/2\} | k \in \{1, \ldots, n\}, p(\mathcal{J}_k) \leq D\}.$$

Given two disjoint sets of jobs $\mathcal{J}_{seq}$ and $\mathcal{J}_D$, we say they are placed *L-shaped*, if the jobs $i \in \mathcal{J}_D$ are placed such that $\sigma(i) + p(i) = D$, while the jobs in $\mathcal{J}_{seq}$ are sorted by energy demand and placed left-aligned most demanding to the left, see Figure 1a.

▶ **Lemma 22.** *Let $T = \max\{T_1, T_2, T_3, T_4\}$. If we place $\mathcal{J}_{\text{seq}} := \mathcal{J}_{e(i)>T/2}$ and $\mathcal{J}_D := \mathcal{J}_{p(i)>D/2} \setminus \mathcal{J}_{\text{seq}}$ L-shaped, the schedule has a height of at most $T + e(\mathcal{J}_D)/2 \leq (3/2)$OPT.*

Proof. Consider a vertical line L through the generated schedule. If L does not intersect a job from $\mathcal{J}_{\text{seq}}$, the intersected jobs have a height of at most $e(\mathcal{J}_D) \leq T$. Otherwise, let $i_L \in \mathcal{J}_{\text{seq}}$ and $\mathcal{J}_{W,L} \subseteq \mathcal{J}_D$ be the jobs intersected by L and define $\mathcal{J}_{\text{seq},L} := \mathcal{J}_{e(i)\geq e(i_L)}$. Note that by definition of the schedule, it holds that $\mathcal{J}_{W,L} \subseteq \mathcal{J}_{p(i)>(\max\{D-p(\mathcal{J}_{\text{seq},L}),D/2\})} \setminus \mathcal{J}_{\text{seq},L}$. Since $e(i_L) > T_4/2$, it holds that $T_4 \geq e(i_L) + e(\mathcal{J}_{W,L})/2$, by definition of T_4. As a consequence $e(i_L) + e(\mathcal{J}_{W,L}) \leq T + e(\mathcal{J}_{W,L})/2 \leq T + e(\mathcal{J}_D)/2 \leq (3/2)$OPT. ◀

B Proof of Theorem 5 (First Steinberg Case)

Proof. We place jobs that are very time consuming or very energy demanding in an ordered fashion, while the residual jobs will be placed using Steinberg's Algorithm, see Figure 1b. We define $\mathcal{J}_D := \mathcal{J}_{p(j)>(1/2+w)D} \setminus \mathcal{J}_{e(j)>T/2}$ to be the set of jobs with large processing times excluding jobs with large energy demands. We place each job $j \in \mathcal{J}_D$ such that $\sigma(j) = D - p(j)$. All the jobs in $\mathcal{J}_{e(j)>T/2}$ are sorted by energy demand and placed left aligned, most demanding first inside the schedule area. Let $\rho := e(\mathcal{J}_D)/T$ and let $e_{(1-2w)D}$ denote the energy demand of the job in $\mathcal{J}_{e(j)>T/2}$ at position $(1-2w)D$. Then $e_{(1-2w)D} \geq (2/3)T$. By Lemma 18 and the choice of T, we know that $e_{(1-2w)D} + e(\mathcal{J}_D) \leq T \leq$ OPT and hence $\rho \leq (1/3)$. Let L be a vertical line though the schedule, that is at or strictly left of $(1/2-w)D$ and intersects a job from $\mathcal{J}_{e(j)>T/2}$ and all the jobs from $\mathcal{J}_D$. By Lemma 22 at and left of L the peak energy demand of the schedule is bounded by $(1+\rho/2)T$. On the other hand, right of L the energy demand of the schedule does not increase compared to L. As a consequence, the peak energy demand in the current schedule is bounded by $(1+\rho/2)T \leq (7/6)T$. Furthermore, we know that right of $(1-2w)D$ the schedule has a peak energy demand of at most T. Consider the set of jobs $\mathcal{J}_{e(j)\in((1/3)T,(1/2)T]}$. By Lemma 15 we know $p(\mathcal{J}_{e(j)\in((1/3)T,(1/2)T]}) \leq 2w \cdot D$, since $\mathcal{J}_{e(j)>(2/3)T} \geq (1-w)D$. Now we consider two cases.

Case A. If $\varepsilon \leq \rho/2$, we place all the jobs in $\mathcal{J}_M := \mathcal{J}_{e(j)\in((1/3)T,(1/2)T]}$ right-aligned next to each other inside the strip. Since they have an energy demand of at most $(1/2)T$ and right of $(1-2w)D$ the schedule has a peak energy demand of at most T, the peak energy demand of $(5/3)T$ is not exceeded after adding these jobs. Define $\lambda := p(\mathcal{J}_M)/D$. Now at each point on the x-axis between 0 and $a := (1-\lambda)D$ the schedule has an energy demand of at most

$(1+\rho/2)T$, and, therefore, we can use an energy demand of $b := (2/3 - \rho/2 + \varepsilon)T$ to place the residual jobs. Let $\mathcal{J}_{res}$ denote the set of residual jobs that still have to be placed. Note that each job in $\mathcal{J}_{res}$ has an energy demand of at most $(1/3)T$ and a processing time of at most $(1/2+w)D$ and the total area of these jobs can be bound by

$$\begin{aligned}\mathrm{work}(\mathcal{J}_{res}) &\leq DT - (2/3)T \cdot (1-w)D - \rho T \cdot (1/2+w)D - (1/3)T \cdot \lambda D\\ &= (1/3 + (2/3)w - \rho(1/2+w) - \lambda/3)DT,\end{aligned}$$

and hence $2\mathrm{work}(\mathcal{J}_{res}) \leq (2/3 + (4/3)w - \rho(1+2w) - (2/3)\lambda)DT$. On the other hand, it holds that

$$\begin{aligned}&ab - (2p_{\max} - a)_+(2e_{\max} - b)_+\\ =&(2/3 - \rho/2 + \varepsilon)T(1-\lambda)D\\ &\quad - ((2(1/2+w) - (1-\lambda))D)_+((2(1/3) - (2/3 - \rho/2 + \varepsilon))T)_+\\ =&(2/3 - \rho/2 + \varepsilon - (2/3)\lambda + (\rho/2 - \varepsilon)\lambda - (2w+\lambda)_+(\rho/2 - \varepsilon)_+)DT\\ =&(2/3 + \varepsilon(1+2w) - (1/2+w)\rho - (2/3)\lambda)DT,\end{aligned}$$

since $\rho/2 - \varepsilon \geq 0$. Hence Steinberg's condition is fulfilled if $(4/3)w - \rho(w+1/2) \leq \varepsilon(1+2w)$, which is true since $w \leq (3/4)\varepsilon$.

Case B. On the other hand, if $\rho/2 < \varepsilon$, it holds that $(2/3 + \varepsilon - \rho/2)/2 \geq 1/3$, and we consider the set $\mathcal{J}_M := \mathcal{J}_{e(j) \in (((2/3+\varepsilon-\rho/2)/2)T,(1/2)T]}$, instead of the set $\mathcal{J}_{e(j) \in ((1/3)T,(1/2)T]}$, and place it right-aligned. Again, we define $\lambda := p(\mathcal{J}_M)$. Now, each job in $\mathcal{J}_{res}$ has an energy demand of at most $(1/3 + \varepsilon/2 - \rho/4)T$ and a processing time of at most $(1/2+w)D$. The total area of these jobs can be bounded by

$$\begin{aligned}\mathrm{work}(\mathcal{J}_{res}) \leq& DT - (2/3)T \cdot (1-w)D - \rho T \cdot (1/2+w)D - (1/3 + \varepsilon/2 - \rho/4)T \cdot \lambda D\\ &= (1/3 + (2/3)w - \rho(1/2+w) - \lambda(1/3 + \varepsilon/2 - \rho/4))DT,\end{aligned}$$

and hence $2\mathrm{work}(\mathcal{J}_{res}) \leq (2/3 + 4/3w - \rho(1+2w) - (2/3 + \varepsilon - \rho/2)\lambda)DT$. On the other hand, it holds that

$$\begin{aligned}&ab - (2p_{\max} - a)_+(2e_{\max} - b)_+\\ &= (2/3 - \rho/2 + \varepsilon)T(1-\lambda)D - (2(1/2+w)D - (1-\lambda)D)_+(2(1/3 + \varepsilon/2 - \rho/4)T\\ &\qquad - (2/3 - \rho/2 + \varepsilon)T)_+\\ &= (2/3 + \varepsilon - \rho/2 - (2/3 - \rho/2 + \varepsilon)\lambda)DT,\end{aligned}$$

Hence, Steinberg's condition is fulfilled if $(4/3 - 2\rho)w - \rho/2 \leq \varepsilon$, which is true since $w \leq (3/4)\varepsilon$.

Therefore, in both cases we use Steinberg's algorithm to place the jobs $\mathcal{J}_{\mathrm{res}}$ inside a rectangular container C of height $(2/3 + \varepsilon - \rho)T$ and width $(1-\lambda)D$, which in turn is positioned at $\sigma(C) = 0$. ◀

C Proof of Theorem 6 (Second Steinberg Case)

Proof. In the first step, we place all the jobs in $\mathcal{J}_D := \mathcal{J}_{p(j)>D/2}$ and $\mathcal{J}_{seq} := \mathcal{J}_{e(j)>T/2} \setminus \mathcal{J}_D$ L-shaped. By Lemma 22 the resulting schedule has a peak energy demand of at most $(3/2)\mathrm{OPT}$. Let $e(\mathcal{J}_D) := (2/3+\rho)T$ and $p(\mathcal{J}_{seq}) := \lambda D$. By Lemma 19, we know that, since $e(\mathcal{J}_{p(j)\geq(3/4)D}) > (2/3)T$, that $\lambda D \leq 2(D - (3/4)D) = D/2$.

The total amount of work $\mathrm{work}(\mathcal{J}_{res})$ of the residual jobs is bounded by

$$\mathrm{work}(\mathcal{J}_{res}) \leq DT - (3/4)D \cdot (2/3)T - (1/2)D \cdot \rho T - \lambda D \cdot (1/2)T = (1/2 - \rho/2 - \lambda/2)DT.$$

On the other hand, there is a rectangular area with time $a := (1-\lambda)D$ and energy $b := ((5/3) - (2/3+\rho))T = (1-\rho)T \geq (1/2)T$ where we can place the residual jobs. We will place the residual jobs into this area using Steinberg's algorithm. This is possible if the Steinberg's condition $2\mathrm{work}(\mathcal{J}_{res}) \leq ab - (2 \cdot p_{\max}(\mathcal{J}_{res}) - a)_+(2 \cdot e_{\max}(\mathcal{J}_{res}) - b)_+$ is fulfilled and each job fits inside the schedule area. Since $p_{\max}(\mathcal{J}_{res}) \leq D/2 \leq a$ and $e_{\max}(\mathcal{J}_{res}) \leq T/2 < b$, it holds that

$$\begin{aligned}
&ab - (2 \cdot p_{\max}(\mathcal{J}_{res}) - a)_+(2 \cdot e_{\max}(\mathcal{J}_{res}) - b)_+\\
&= (1-\lambda)D \cdot (1-\rho)T - (D - (1-\lambda)D)_+(T - (1-\rho)T)_+\\
&= (1-\lambda-\rho)D \cdot T\\
&= 2(1/2 - \rho/2 - \lambda/2)DT \geq 2\mathrm{work}(\mathcal{J}_{res}).
\end{aligned}$$

The condition is fulfilled, and we can use the free rectangular area to place the residual jobs. ◀