Don't Rock the Boat: Algorithms for Balanced Dynamic Loading and Unloading*

Sándor P. Fekete[1], Sven von Höveling[1], Joseph S. B. Mitchell[†2], Christian Rieck[1], Christian Scheffer[1], Arne Schmidt[1], and James R. Zuber[2]

[1]Department of Computer Science, TU Braunschweig, 38106 Braunschweig, Germany. {s.fekete,v.sven,c.rieck,c.scheffer,arne.schmidt}@tu-bs.de
[2]Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794, USA. joseph.mitchell@stonybrook.edu, zuber139@gmail.com

Abstract

We consider dynamic loading and unloading problems for heavy geometric objects. The challenge is to maintain balanced configurations at all times: minimize the maximal motion of the overall center of gravity. While this problem has been studied from an algorithmic point of view, previous work only focuses on balancing the *final* center of gravity; we give a variety of results for computing balanced loading and unloading schemes that minimize the maximal motion of the center of gravity during the entire process.

In particular, we consider the one-dimensional case and distinguish between *loading* and *unloading*. In the unloading variant, the positions of the intervals are given, and we search for an optimal unloading order of the intervals. We prove that the unloading variant is NP-complete and give a 2.7-approximation algorithm. In the loading variant, we have to compute both the positions of the intervals and their loading order. We give optimal approaches for several variants that model different loading scenarios that may arise, e.g., in the loading of a transport ship with containers.

1 Introduction

Packing a set of objects is a classic challenge that has been studied from a wide range of angles: how can the objects be arranged to fit into the container? Packing problems are important for a large spectrum of practical applications, such as loading items into a storage space, or containers onto a ship. They are also closely related to problems of scheduling and sequencing, in which the issues of limited space are amplified by including temporal considerations.

Packing and scheduling are closely intertwined in *loading* and *unloading* problems, where the challenge is not just to compute an acceptable *final* configuration, but also the process of *dynamically building* this configuration, such that intermediate states are both achievable and stable. This is highly relavant in the scenario of loading and unloading container ships, for which maintaining *balance* throughout the process is crucial. Balancedness of packing also plays an important role for other forms of shipping: Mongeau and Bes [13] showed that displacing the center of gravity by less than 75cm in a long-range aircraft may cause, over a 10,000 km flight, an additional consumption of 4,000 kg of fuel.

In this paper, we consider algorithmic problems of balanced loading and unloading. For unloading, this means planning an optimal sequence for removing a given set of objects, one at a time; for loading, this requires planning both position and order of the objects.

The practical constraints of loading and unloading motivate a spectrum of relevant scenarios. As ships are symmetric around their main axis, we focus on one-dimensional settings, in which the objects correspond to intervals. Containers may be of uniform size, but stackable up to a certain limited height;

*This is the full version of an extended abstract that will appear in the 13th Latin American Theoretical INformatics Symposium (LATIN 2018), April 16–19, 2018.
†Work of this author is partially supported by the National Science Foundation (CCF-1526406).

Figure 1: Loading and unloading container ships.

because sliding objects on a moving ship are major safety hazards, stability considerations may prohibit gaps between containers. On the other hand, containers of extremely different size pose particularly problematic scenarios, which is why we also provide results for sets of containers whose sizes are exponentially growing.

1.1 Our Contributions

Our results are as follows; throughout the paper, *items* are the objects that need to be loaded (also sometimes called *placed*) or unloaded, while *container* refers to the space that accomodates them. Furthermore, we assume all objects to have unit density, i.e., their weights correspond to their lengths. In most cases, items correspond to geometric intervals.

- For unloading, we show that it is NP-complete to compute an optimal sequence. More formally, given a set of placed intervals $\{I_1, \ldots, I_n\}$, it is NP-complete to compute an order $\langle I_{\pi(1)}, \ldots, I_{\pi(n)} \rangle$, in which intervals are removed one at a time, such that the maximal deviation of the gravity's center is minimized.
- We provide a corresponding polynomial-time 2.7-approximation algorithm. In particular, we give an algorithm that computes an order of the input intervals such that removing the intervals in that order results in a maximal deviation which is no larger than 2.7 times the maximal deviation induced by an optimal order.
- For loading, we give a polynomial-time algorithm for the setting in which gaps are not allowed. In particular, given a set of lengths values $\ell_1, \ldots, \ell_n \in \mathbb{R}_{>0}$, we require a sequence $\langle I_{\pi(1)}, \ldots, I_{\pi(n)} \rangle$ of pairwise disjoint intervals with $|I_{\pi(i)}| = \ell_{\pi(i)}$ for $i = 1, \ldots, n$ such that the following holds: Placing the interval $I_{\pi(i)}$ in the i-th step results in an n-stepped loading process such that the union of the loaded intervals is connected for all points in time during the loading process. Among these *connected placements*, we compute one for which the maximal deviation of the center of gravity is minimized.
- We give a polynomial-time algorithm for the case of stackable unit intervals. More formally, given an input integer $\mu \geqslant 1$, in the context of the previous variant, we relax the requirement that the union of the placed intervals has to be connected and additionally allow that the placed intervals may be stacked up to a height of μ in a stable manner, defined as follows. We say that layer 0 is completely *covered*. An interval I can be placed, i.e., covered, in layer $k \geqslant 1$ if the interval I is covered in all layers $0, \ldots, k-1$ and if I does not overlap with another interval already placed in layer k.
- We give a polynomial-time algorithm for the case of exponentially growing lengths. More formally, in the context of the previous variant, we require that all intervals are placed in layer 1 and assume that the lengths of the input intervals' lengths are exponentially increasing, i.e., there is an $x \geqslant 2$ such that $x \cdot \ell_i = \ell_{i+1}$ holds for all $i \in \{1, \ldots, n-1\}$.

1.2 Related Work

Previous work on cargo loading covers a wide range of specific aspects, constraints and objectives. The general Cargo Loading Problem (CLP) asks for an optimal packing of (possibly heterogeneous) rectangular boxes into a given bin, equivalent to the Cutting Stock Problem [9]. Most of the proposed methods are heuristics based on (mixed) integer programming and have been studied both for heterogeneous and homogeneous items. Bischoff and Marriott [2] show that the performances of some heuristics may depend on the kind of cargo.

Amiouny et al. [1] consider the problem of packing a set of one-dimensional boxes of different weights and different lengths into a flat bin (so they are not allowed to stack these boxes), in such a way that after placing the last box, the center of gravity is as close as possible to a fixed target point. They prove strong NP-completeness by a reduction from 3-Partition and give a heuristic with a guaranteed accuracy within $\ell_{max}/2$ of a given target point, where ℓ_{max} is the largest box length. A similar heuristic is given by Mathur [12].

Gehring et al. [8] consider the general CLP, in which (rectangular) items may be stacked and placed in any possible position. They construct non-intersecting *walls*, i.e. packings made from similar items for slices of the original container, to generate the overall packing. They also show that this achieves a good final balancing of the loaded items. Mongeau and Bes [13] consider a similar variant in which the objective is to maximize the loaded weight. In addition, there may be other paramaters, e.g., each item may have a different priority [21]. A mixed integer programming approach on this variant is given by Vancroonenburg et al. [22]. Limbourg et al. [10] consider the CLP based on the moment of inertia. Gehring and Bortfeldt [7] give a genetic algorithm for *stable* packings. Fasano [6] considers packing problems of three-dimensional *tetris*-like items in combination with balancing constraints. His work is done within the context of the Automated Transfer Vehicle, which was the European Space Agency's transportation system supporting the International Space Station (ISS).

Another variant is to consider multiple *drops*, for which loaded items have to be available at each drop-off point in such a way that a rearrangement of the other items is not required; see e.g. [3], [4], and [11]. Davies and Bischoff [5] propose an approach that produces a high space utilization for even weight distribution. These scenarios often occur in container loading for trucks, for which the objective is to achieve an even weight distribution between the axles. For a state-of-the-art survey of vehicle routing with different loading constraints and a spectrum of scenarios, see Pollaris et al. [18].

In the context of distributing cargo by sea, two different kind of ships are distinguished: *Ro-Ro* and *Lo-Lo* ships. Ro-Ro (for roll on–roll off) ships carry wheeled cargo, such as cars and trucks, which are driven on and off the ship. Some approaches and problem variants such as multiple drops, additional loading, and optional cargo as well as routing and scheduling considering Ro-Ro ships are considered by Øvstebø et al. [14, 15]. On the other hand, Lo-Lo (load on–load off) ships are cargo ships that are loaded and unloaded by cranes, so any feasible position can be directly reached from above.

While all of this work is related to our problem, it differs in not requiring the center of gravity to be under control for each step of the loading or unloading process. A problem in which a constraint is imposed at each step of a process is Compact Vector Summation (CVS), which asks for a permutation of a set of k-dimensional vectors in order to control their sum, keeping each partial sum within a bounded k-dimensional ball. See Sevastianov [19, 20] for a summary of results in CVS and its application in job scheduling.

A different, but related stacking problem is considered by Paterson and Zwick [17] and Paterson et al. [16], who consider maximizing the reach beyond the edge of a table by stacking n identical, homogeneous, frictionless bricks of length 1 without toppling over, corresponding to keeping the center of gravity of subarrangements supported.

2 Preliminaries

An *item* is a unit interval $I := [m - \frac{1}{2}, m + \frac{1}{2}]$ with midpoint m. A set $\{I_1, \ldots, I_n\}$ of n items with midpoints $m_1, \ldots, m_n$ is *valid* if $m_i = m_j$ or $|m_i - m_j| \geqslant 1$ holds for all $i, j = 1, \ldots, n$. The *center of gravity* $C(I_1, \ldots, I_n)$ of a valid set $\{I_1, \ldots, I_n\}$ of items is defined as $\frac{1}{n}\sum_{i=1}^{n} m_i$.

Given a valid set $\{I_1, \ldots, I_n\}$ of items, we seek orderings in which each item I_j is removed or placed such that the maximal deviation for all points in time $j = 1, \ldots, n$ is minimized. More formally, for $j =$

$1, \ldots, n$ and a permutation $\pi : j \mapsto \pi_j$, let $C_j := C\left(I_{\pi_j}, \ldots, I_{\pi_n}\right)$. The Unloading Problem (Unload) seeks to minimize the maximal deviation during an unloading process of $I_1, \ldots, I_n$. In particular, given an input set $\{I_1, \ldots, I_n\}$ of items, we seek a permutation π such that $\max_{i,j=1,\ldots,n} |C_i - C_j|$ is minimized. (Equivalently, the Unload can be posed as a *loading* problem in which the positions of the items is given, and we seek an order of loading them.)

In the Loading Problem (Load) we relax the constraint that the positions of the considered items are part of the input. In particular, we seek an ordering and a set of midpoints for the containers such that the containers are disjoint and the maximal deviation for all points in time of the loading process is minimized; see Section 4 for a formal definition.

3 Unloading

We show that the problem Unload is NP-complete and give a polynomial-time 2.7-approximation algorithm for Unload. We first show that there is a polynomial- time reduction from the discrete version of Unload, the Discrete Unloading Problem (dUnload), to Unload; this leads to a proof that Unload is NP-complete, followed by a 2.7-approximation algorithm for Unload.

In the Discrete Unloading Problem (dUnload), we do not consider a set of items, i.e., unit intervals, but a discrete set $X := \{x_1, \ldots, x_n\}$ of points. The center of gravity $C(X)$ of X is defined as $\frac{1}{n}\sum_{i=1}^{n} x_i$. For $j = 1, \ldots, n$ and a permutation $\pi : j \mapsto \pi_j$, let $C_j = C\left(x_{\pi_j}, \ldots, x_{\pi_n}\right)$. Again, we seek a permutation such that $\max_{i,j=1,\ldots,n} |C_i - C_j|$ is minimized.

Lemma 1. *There is a polynomial-time reduction from* Unload *to* dUnload*.*

Proof. Consider the items $I_1, \ldots, I_n$ with their midpoints $m_1, \ldots, m_n$. We choose $x_i = m_i$ to get a discrete set of points. It is easy to see that the center of gravity does not change, i.e., after removing k intervals, and for any permutation π, we have $\frac{1}{k+1}\sum_{i=n-k}^{n} x_{\pi_i} = \frac{1}{k+1}\sum_{i=n-k}^{n} m_{\pi_i}$. Because this holds for any k and π we conclude that an optimal solution to dUnload is an optimal solution to Unload. □

Lemma 2. *There is a polynomial-time reduction from* dUnload *to* Unload*.*

Proof. Consider a set X of points and the smallest distance d among all pairs of points. We construct unit intervals $I_1, \ldots, I_n$ with midpoints $m_i = \frac{x_i}{d}$. After removing k intervals, and for any permutation π, the center of gravity is at position $\frac{1}{k+1}\sum_{i=n-k}^{n} m_{\pi_i} = \frac{1}{d(k+1)}\sum_{i=n-k}^{n} x_{\pi_i}$, i.e., the center of gravity is scaled by a factor of d. Because this holds for any k and π we conclude that an optimal solution to Unload is an optimal solution to dUnload. □

The combination of Lemma 1 and Lemma 2 implies Corollary 3.

Corollary 3. Unload *and* dUnload *are polynomial-time equivalent.*

3.1 NP-Completeness of the Discrete Case

We can establish NP-completeness of the discrete problem dUnload.

Theorem 4. dUnload *is* NP*-complete.*

Proof. Our reduction is from 3-Partition. An instance of 3-Partition takes as input a multiset $Y = \{y_1, y_2, \ldots, y_{3m}\}$ of $3m$ positive integers and asks if it is possible to partition Y into m triples, $Y_1, Y_2, \ldots, Y_m$, such that the sum of the integers in each triple Y_i is exactly $B = (\sum_{j=1}^{3m} y_j)/m$. The problem 3-Partition is known to be NP-complete, and it remains NP-complete when one assumes that $B/4 < y_j < B/2$, for all $j = 1, 2, \ldots, 3m$. Given such an instance, $Y = \{y_1, y_2, \ldots, y_{3m}\}$, we construct an instance of dUnload such that the set Y has the desired partition into m triplets if and only if the instance of dUnload has a *feasible ordering*, for which the center of gravity is always within the interval $[0, B/M]$.

Our instance consists of the following set $X = \{x_1, x_2, \ldots\}$ of points:

(i) M points at the origin, 0, for a large integer $M > m$ to be specified below (we see that $M > 4mB$ suffices).

(ii) $3m$ points at the $3m$ integers $y_i \in Y$.

(iii) m points at $-B$.

In the following basic description, this instance X of dUnload is a multiset, i.e., it has repeated points: multiple points at 0, at $-B$, and potentially at points of Y, so it forms a multiset. These points can be perturbed to be distinct, and then the instance can be rescaled so that the minimum distance between consecutive points is at least 1.

We let $\sigma_k = x_{\pi_1} + \cdots x_{\pi_k}$ denote the partial sum of the first k points in the ordered sequence. Our goal is to decide if there exists an ordering π of the $n = 3m + m + M = |X|$ points of X so that the centers of gravity, $C_k = C(x_{\pi_1}, \ldots, x_{\pi_k})$, remain within the interval $[0, B/M]$, for all $k = 1, 2, \ldots, n$. (For simplicity our description here is in terms of adding the points one by one; for the unloading order, time is reversed.)

The center of gravity of all $n = 4m + M$ of the points of X is $C_n = \frac{M \cdot 0 + Bm + m \cdot (-B)}{4m+M} = 0$.

We claim that the centers of gravity C_k will lie within the interval $[0, B/M]$, for all $k = 1, \ldots, n$, if and only if the points X are ordered as follows. First, all M of the points of type (i) at the origin, then three points (in any order) of type (ii) that sum to exactly B, then one point at $-B$ of type (iii), then three points of type (ii) that sum to exactly B, then one point of type (iii), etc. This means that the centers of gravity C_k will lie within the interval $[0, B/M]$, for all $k = 1, \ldots, n$, if and only if there is a partition of Y into triples, each summing to exactly B.

The "if" direction of the claim is straightforward: If Y has a partition into triples, each summing to exactly B, then the specified ordering of the points X (M points at 0, followed by a triple that sums to exactly B, followed by a point at $-B$, followed by another triple that sums to exactly B, etc) achieves the desired containment of the center of gravity, C_k, in the interval $[0, B/M]$, for each k.

For the "only if" claim, first note that if the sequence does not begin with M elements all at the origin, having instead a non-zero element at position $k < M$, then the center of gravity C_k is either $-B/k < 0$ or is at least $1/k > 1/M$, because each of the type (ii) points is a positive integer from the set Y. Thus, a feasible sequence π (that maintains the center of gravity within $[0, B/M]$) must begin with M elements of type (i), namely points at the origin. If the next point is of type (iii), at $-B$, then the center of gravity becomes negative, and falls outside of $[0, B/M]$; thus, the $(M + 1)$st element must be a positive integer. The partial sum $\sigma_k = x_{\pi_1} + \cdots x_{\pi_k}$ of the first k points in the sequence must never become negative (or else the center of gravity, σ_k/k, will fall below the target interval $[0, B/M]$) and must never become greater than B, or else the center of gravity will fall above the target interval $[0, B/M]$, as we now argue. If $\sigma_k > B$, then $\sigma_k \geqslant B + 1$, making the center of gravity at least $\frac{B+1}{k}$. Now if we pick M to be large enough (it suffices to pick $M > 4mB$), then $\frac{B+1}{k} > \frac{B}{M}$, for all $k = 1, 2, \ldots, M + 4m = n$, as claimed. Thus, in order for the center of gravity C_k to remain in $[0, B/M]$ for all k, the partial sum must never get above B and must never become negative. The only way this can be accomplished is to sequence the points of types (ii) and (iii) as claimed, into triples of points of Y that sum to exactly B, followed by a point $-B$, followed by another triple of points of Y that sum to exactly B, etc. We conclude that if the instance X has a solution, keeping the center of gravity within the target interval $[0, B/M]$, then the input instance of 3-Partition, Y, has a solution. □

Because of the polynomial-time equivalance of dUnload and Unload, we conclude the following.

Corollary 5. Unload *is* NP*-complete.*

3.2 Lower Bounds and an Approximation Algorithm

When unloading a set of items, their positions are fixed, so (after reversing time) unloading is equivalent to a loading problem with predetermined positions. For easier and uniform notation throughout the paper, we use this latter description.

In order to develop and prove an approximation algorithm for dUnload, we begin by examining lower bounds on the span, $R - L$, of a minimal interval, $[L, R]$, containing the centers of gravity at all stages in an optimal solution.

Without loss of generality, we assume that the input points x_i sum to 0 (i.e., $\sum_i x_i = 0$), so that the center of gravity, C_n, of all n input points is at the origin. We let $R = \max_i C_i$ and $L = \min_i C_i$. Our first simple lemma leads to a first (fairly weak) bound on the span.

Lemma 6. *Let $(x_1, x_2, x_3, \ldots)$ be any sequence of real numbers, with $\sum_i x_i = 0$. Let $C_j = (\sum_{i=1}^{j} x_i)/j$ be the center of gravity of the first j numbers, and let $R = \max_i C_i$ and $L = \min_i C_i$. Then, $|R - L| \geqslant \frac{|x_i|}{i}$, for all $i = 1, 2, \ldots$.*

Proof. We see that

$$C_i = \frac{x_i + \sum_{j=1}^{i-1} x_j}{i} = \frac{i-1}{i} C_{i-1} + \frac{x_i}{i};$$

thus,

$$C_i - C_{i-1} = \frac{x_i - C_{i-1}}{i}.$$

We now distinguish two cases. If x_i and C_{i-1} have opposite signs, then we get

$$|C_i - C_{i-1}| = \frac{|x_i - C_{i-1}|}{i} = \frac{|x_i| + |C_{i-1}|}{i} \geqslant |\frac{x_i}{i}|,$$

implying that

$$|R - L| \geqslant |C_i - C_{i-1}| \geqslant \frac{|x_i|}{i},$$

because the interval $[L, R]$ must be large enough to contain any amount of change, $|C_i - C_{i-1}|$, to the center of gravity, at any step i. On the other hand, if x_i and C_{i-1} have the same signs (i.e., $\mathrm{sign}(C_{i-1}x_i) = 1$), then x_i, C_i, and C_{i-1} all have the same signs, and we get

$$|C_i| = \frac{i-1}{i}|C_{i-1}| + \frac{|x_i|}{i} \geqslant \frac{|x_i|}{i},$$

implying that

$$|R - L| \geqslant |C_i| \geqslant \frac{|x_i|}{i},$$

because the interval $[L, R]$ must be large enough to contain both C_i and the overall center of gravity, 0, for any i. Thus, in all cases, $|R - L| \geqslant \frac{|x_i|}{i}$. □

Corollary 7. *For any valid solution to* dUnload*, the minimal interval $[L, R]$ containing the center of gravity at every stage must have length $|R - L| \geqslant \frac{|u_i|}{i}$ where u_i is the input point with the i-th smallest magnitude.*

We note that the naive lower bound given by Corollary 7 can be far from tight: Consider the sequence $1, 2, 3, 4, 5, 6, 7, -7, -7, -7, -7$. In the optimal order, the first -7 is placed fourth, after $2, 1, 3$. The optimal third and fourth centers, $\{2, -\frac{1}{4}\}$ are the largest magnitude positive and negative centers seen, and show a span 2.25 times greater than the naive bound of 1. By placing the first -7 in the third position, $R \geqslant \frac{3}{2}$, and $L \leqslant -\frac{4}{3}$. By placing it fifth, $R \geqslant \frac{5}{2}$. Our observation was that failing to place our first -7 if the cumulative sum is > 7 would needlessly increase the span.

This generalizes to the sequence $(x_1 = 1, x_2 = 2, \ldots, x_{k-1} = k-1, x_k = -k, x_{k+1} = -(k+1), \ldots, x_N)$, with an appropriate x_N to make $\sum x_i = 0$. If we place positive weights in increasing order until $c_l \geqslant \frac{k}{l}$, placing $-k$ instead of a positive at position l would decrease the center of gravity well below $\frac{k}{l}$. The first negative should be placed when $\min_l \frac{l^2 - l}{2} \geqslant k$, which is when $l \approx \sqrt{2k}$. In this example, our optimal center of gravity span is at least $\frac{k}{l} \approx \sqrt{\frac{k}{2}}$, not the 1 from the naive bound of Corollary 7.

We now describe our heuristic, $\mathcal{H}$, which leads to a provable approximation algorithm. It is convenient to relabel and reindex the input points as follows. Let $(P_1, P_2, \ldots)$ denote the positive input points, ordered (and indexed) by increasing value. Similarly, let $(N_1, N_2, \ldots)$ denote the negative input points, orders (and indexed) by increasing magnitude $|N_i|$ (i.e., ordered by decreasing value).

The heuristic $\mathcal{H}$ orders the input points as follows. The first point is simply the one closest to the origin (i.e., of smallest absolute value). Then, at each step of the algorithm, we select the next point in the order by examining three numbers: the partial sum, S, of all points placed in the sequence so far, the smallest magnitude point, α, not yet placed that has the same sign as S, and the smallest magnitude point, β, not yet placed that has the opposite sign of S. If $S + \alpha + \beta$ is of the same sign as S, then we place β next in the sequence; otherwise, if $S + \alpha + \beta$ has the opposite sign as S, then we place α next in the sequence. The intuition is that we seek to avoid the partial sum from drifting in one direction; we switch to the opposite sign sequence of input points in order to control the drift, when it becomes expedient to do so, measured by comparing the sign of S with the sign of $S + \alpha + \beta$, where α and β are the smallest magnitude points available in each of the two directions. We call the resulting ordering the $\mathcal{H}$-permutation. The $\mathcal{H}$-permutation puts the j-th largest positive point, P_j, in position π_j^+ in the order, and puts the j-th largest in magnitude negative point, N_j, in position π_j^- in the order, where

$$\pi_j^+ = j + \max_k \{k \ : \ \sum_{i=1}^{k} |N_i| \leqslant \sum_{l=1}^{j} P_l\} \quad \text{and} \quad \pi_j^- = j + \max_k \{k \ : \ \sum_{i=1}^{k} P_i < \sum_{l=1}^{j} |N_l|\}.$$

We obtain an improved lower bound based on our heuristic, $\mathcal{H}$, which orders the input points according to the $\mathcal{H}$-permutation.

Lemma 8. *A lower bound on the optimal span of* dUnload *is given by* $|R-L| \geqslant \frac{P_i}{\pi_i^+}$ *and* $|R-L| \geqslant \frac{|N_i|}{\pi_i^-}$.

To prove the lemma, we begin with a claim.

Claim 9. *For any input set to the discrete unloading problem, where s_i are all terms with the same sign sorted by magnitude, a permutation π that minimizes the maximum value of the ratio $\frac{|s_i|}{\pi_i}$ must satisfy $\pi_k < \pi_i$, for all $k < i$.*

Proof. By contradiction, assume that the minimizing permutation π has the maximum value of the ratio $\frac{|s_i|}{\pi_i}$ occur at an i for which there exists a $k < i$ for which $\pi_i \leqslant \pi_k$, which means that $\pi_i < \pi_k$ (because π_i cannot equal π_k for a permutation π, and $k \neq i$).

Because the terms s_i are indexed in order sorted by magnitude, $|s_k| \leqslant |s_i|$. Exchanging the order of s_i and s_k in the permutation would lead to two new ratios in our sequence: $\frac{|s_i|}{\pi_k}$ and $\frac{|s_k|}{\pi_i}$. Because $\pi_k > \pi_i$, we get $\frac{|s_i|}{\pi_k} < \frac{|s_i|}{\pi_i}$. Because $|s_k| \leqslant |s_i|$, we get $\frac{|s_k|}{\pi_i} \leqslant \frac{|s_i|}{\pi_i}$. Because these new ratios are smaller than $\frac{|s_i|}{\pi_i}$, we get a contradiction to the fact that π minimizes the maximum ratio. □

The following claim is an immediate consequence of Lemma 6.

Claim 10. *For the i maximizing $\frac{P_i}{\pi_i^+}$, any ordering placing this element earlier than π_i^+ in the sequence has a span $|R - L| > \frac{P_i}{\pi_i^+}$. Similarly, for the i maximizing $\frac{|N_i|}{\pi_i^+}$, any ordering placing this element earlier than π_i^- in the sequence has a span $|R - L| > \frac{|N_i|}{\pi_i^-}$.*

On the other hand, the following holds.

Claim 11. *For the i maximizing $\frac{P_i}{\pi_i^+}$, any ordering placing this element later than π_i^+ in the sequence has a span $|R - L| > \frac{P_i}{\pi_i^+}$. A similar statement holds for $\frac{|N_i|}{\pi_i^-}$.*

Proof. The proof is by contradiction. The index into the $\mathcal{H}$ permutation maximizing the ratio $\frac{|x_k|}{k}$ is i. We assume (wlog) $x_i = P_J > 0$, and we let $K = i - J$.

If P_J is not placed in position i, we suppose another element, x, can be placed in its stead and results in a span that is less than $\frac{P_J}{i}$.

When placing any positive $x > P_J$ in the initial i position, the lowest possible observed span from Lemma 6 is at least $\frac{x}{i} > \frac{P_J}{i}$, which would contradict our assumption. Similarly, all positive points placed before or at position i must be less than or equal to P_J.

All permutations of these $J-1$ positive elements and the first $K+1$ negative elements have a large negative center of gravity at position i. From $K = \max_k\{k \ : \ \sum_{i=1}^{k} |N_i| \leqslant \sum_{l=1}^{J} P_l\}$, we get $\sum_{i=1}^{K+1} |N_i| \geqslant \sum_{l=1}^{J} P_l$, and hence $\sum_{i=1}^{K+1} N_i + \sum_{l=1}^{J} P_l \leqslant 0$, implying $\sum_{i=1}^{K+1} N_i + \sum_{l=1}^{J-1} P_l \leqslant -P_J$. Therefore, the maximizing value satisfies

$$|c^*| = \frac{|\sum_{i=1}^{K+1} N_i + \sum_{l=1}^{J-1} P_l|}{i} \geqslant \frac{P_J}{i}$$

Because the center of gravity is at a location greater than the $\mathcal{H}$-bound, and $R \geqslant 0 \geqslant L$, this span is also greater than the $\mathcal{H}$-bound and we can neither place an element greater than P_J nor one less than P_J in place of P_J while lowering the span beneath the $\mathcal{H}$-bound. □

Theorem 12. *The $\mathcal{H}$-permutation minimizes the maximum (over i) value of the ratio $\frac{|x_i|}{\pi_i}$, and thus yields a lower bound on $|R - L|$.*

For the worst-case ratio, we get the following.

Theorem 13. *The $\mathcal{H}$ heuristic yields an ordering having span $R - L$ at most 2.7 times larger than the $\mathcal{H}$-lower bound.*

Proof. Before separating the input points into the sorted P and N lists, we normalize their values so that the maximum value of the ratio $\frac{|x_i|}{i}$ is 1. This implies that $|x_i| \leqslant i$, for all i.

When using the $\mathcal{H}$-permutation, whenever we place an element of opposite sign from the current center of gravity, C_i, we know that the partial sum S_i and center of gravity C_i obey $|S_i| \leqslant |x_{i+1}|$. Using normalization, we have $|x_{i+1}| \leqslant i + 1$, hence $|C_i| = |\frac{S_i}{i}| \leqslant \frac{i+1}{i}$.

When the center of gravity reaches its leftmost extent, we cannot place another negative element, because the next largest negative element would push the center of gravity further to the left. A similar statement holds for the rightmost extent and positive elements. This means that if the center of gravity first reaches L at step a and first reaches R at step b, then $L \geqslant \frac{-(a+1)}{a}$ and $R \leqslant \frac{b+1}{b}$ (*), so $R-L \leqslant 2+\frac{1}{a}+\frac{1}{b}$ (**).

The final step is to argue that the right-hand side is bounded by 2.7.

Let us assume $a < b$ and consider the ratio $\alpha(a, b)$ of the span we obtain from $\mathcal{H}$ to the $\mathcal{H}$-bound. We consider small values of a and b and can provide the following bounds.

$\alpha(\mathbf{1}, \mathbf{b}) \leqslant \mathbf{2.5}$. This holds as follows. We have $L = \frac{N_1}{1} \geqslant \frac{-1}{1}$ by normalization of masses. As $b \geqslant 2$, it follows from Equation (*) that $R \leqslant \frac{3}{2}$. Thus, $R - L \leqslant 2\frac{1}{2}$.

$\alpha(\mathbf{2}, \mathbf{3}) \leqslant \mathbf{1.5}$. This follows from considering the first terms: We observe that $L = \frac{N_1+N_2}{2}$ and $R = \frac{N_1+N_2+P_1}{3}$. Moreover, $P_1 \leqslant 3$ by normalization of masses and $|N_1 + N_2| \leqslant P_1$ from the $\mathcal{H}$-ordering on P_1. Therefore, $R - L = \frac{2P_1-N_1-N_2}{6} \leqslant \frac{3P_1}{6} \leqslant \frac{3}{2}$.

$\alpha(\mathbf{2}, \mathbf{4}) \leqslant \mathbf{2.5}$. Again observe that $L = \frac{N_1+N_2}{2}$, as well as $R = \frac{N_1+N_2+P_1+P_2}{4}$, so $R-L = \frac{P_1+P_2-N_1-N_2}{4}$. By $\mathcal{H}$-ordering, we get $|N_1 + N_2| \leqslant P_1$, so $R - L \leqslant \frac{2P_1+P_2}{4}$. Again by normalization of masses, we have $P_1 \leqslant 3$ and $P_2 \leqslant 4$, implying $R - L \leqslant \frac{10}{4} = 2.5$

Therefore, we only have to consider $\alpha(2, b \geqslant 5) \leqslant 2.7$, which follows from Equation (**). □

Corollary 14. *There is a polynomial-time* 2.7*-approximation algorithm for* Unload.

4 Loading

We now consider loading problems, for which we require some additional definitions: The positions of the objects are part of the optimization, and, for some loading variants, the items may have different lengths. Consider the following more general definitions.

An *item* is given by a real number ℓ. By assigning a *position* $m \in \mathbb{R}$ to an item, we obtain an interval I with length ℓ and midpoint m. For $n \geqslant 1$, we consider a set $\{\ell_1, \dots, \ell_n\}$ of n items and assume $\ell_1 \geqslant \dots \geqslant \ell_n$. Furthermore, $\{\ell_1, \dots, \ell_n\}$ is *uniform* if $\ell := \ell_1 = ... = \ell_n$.

A *state* is a set $\{(I_1, h_1), \dots, (I_n, h_n)\}$ of pairs, each one consisting of an interval I_j and an integer $h_j \geqslant 1$, the *layer* in which I_j lies. A state satisfies the following: (1) Two different intervals that lie in the same layer do not overlap and (2) for $j = 2, \dots, n$, an interval in layer j is a subset of the union of the intervals in layer $j - 1$.

A state $\{(I_1, h_1), \dots, (I_n, h_n)\}$ is *plane* if all intervals lie in the first layer.

To simplify the following notations, we let m_j denote the midpoint of the interval I_j, for $j = \{1, \dots, n\}$. The *center of gravity* $C(s)$ of a state $s = \{(I_1, h_1), \dots, (I_n, h_n)\}$ is defined as $\frac{1}{M}\sum_{j=1}^{n} \ell_j m_j$, where M is defined as $\sum_{j=1}^{n} \ell_j$.

A *placement* p of an n-system S is a sequence $\langle I_1, \dots, I_n \rangle$ such that $\{(I_1, h_1), \dots, (I_j, h_j)\}$ is a state, the *j-th state* s_j, for each $j = 1, \dots, n$. The 0-th state s_0 is defined as $\varnothing$ and its center of gravity $C(s_0)$ is defined as 0.

Definition 15. *The* loading problem *(*Load*) is defined as follows: Given a set of n items, determine a placement p such that the $n + 1$ centers of gravity of the $n + 1$ states of p lie close to* 0*. In particular,* *the* deviation $\Delta(p)$ *of a placement* p *is defined as* $\max_{j=0,\ldots,n} |C(s_j)|$. *We seek a placement of* S *with minimal deviation among all possible placements for* S.

We say that stacking is not allowed *if we require that all intervals are placed in layer* 1. *Otherwise, we say that* stacking is allowed. *For a given integer* $\mu \geqslant 1$ *we say that* μ *is the* maximum stackable height *if we require that all used layers are no larger than* μ.

Note that in the loading case, minimizing the deviation is equivalent to minimizing the diameter, i.e., minimizing the maximal distance between the smallest and largest extent of the centers.

4.1 Optimally Loading Unit Items With Stacking

Now we consider the case of unit items for which stacking is allowed. We give an algorithm that optimally loads a set of unit items with stacking.

Theorem 16. *There is a polynomial-time algorithm for loading a set of unit items so that the deviation of the center of gravity is in* $[0, \frac{1}{1+\mu}]$*, where* μ *is the maximum stackable height.*

Proof. Let m_i be the midpoint of item ℓ_i. Because we are allowed to stack items up to height μ, the strategy is the following: set $m_1 = m_2 = \cdots = m_\mu = \frac{1}{1+\mu}$, i.e., the first μ items are placed at the very same position. Call these first μ items the *starting stack* $\mathcal{S}_0$. Subsequently, we place the following items on alternating sides of $\mathcal{S}_0$, i.e., the item $\ell_{\mu+1}$ is placed as close as possible on the left side of $\mathcal{S}_0$, $\ell_{\mu+2}$ is placed as close as possible on the right side, $\ell_{\mu+3}$ is placed on top of $\ell_{\mu+1}$ (if we did not already reach the maximum stackable height of μ), or next to $\ell_{\mu+1}$ (if $\ell_{\mu+1}$ is on the μ-th layer), etc.

After each placement of $\ell_i, 1 \leqslant i \leqslant \mu$, we have $C(\ell_i) = \frac{1}{1+\mu}$. After two more placed items, the center of gravity is again at $\frac{1}{1+\mu}$, because these items neutralize each other. Thus, the critical part is a placement on the left side of $\mathcal{S}_0$. We proceed to show that after placing an item on the left side, the center of gravity is at position at least 0.

The midpoint $m_{\mu+1}$ of the item $\ell_{\mu+1}$ is $\frac{-\mu}{1+\mu}$, thus $C(\ell_{\mu+1}) = \frac{\mu}{1+\mu} - \frac{\mu}{1+\mu} = 0$. Now assume that we have already placed $c = (2k+1)\cdot\mu + \zeta$ items, where $\zeta < 2\mu$ and odd, i.e., we have already placed the starting stack $\mathcal{S}_0$ and k additional stacks of height μ on each side of $\mathcal{S}_0$. Let $z := (2k+1)\cdot\mu$. Then the center of gravity is at position $C(c)$, where

$$\begin{aligned} C(c) =& \frac{z\cdot\frac{1}{1+\mu} + \sum\limits_{i=z+1}^{z+\zeta} m_i}{z+\zeta} = \frac{(z+\zeta-1)\cdot\frac{1}{1+\mu} + \frac{-k\mu-k-\mu}{1+\mu}}{z+\zeta} = \frac{k\mu+\zeta-1-k}{(1+\mu)\cdot(z+\zeta)} \\ =& \frac{k(\mu-1)+\zeta-1}{(1+\mu)\cdot(z+\zeta)} \geqslant \frac{\zeta-1}{(1+\mu)\cdot(z+\zeta)} \geqslant \frac{0}{(1+\mu)\cdot(z+\zeta)} \geqslant 0. \end{aligned}$$

□

In the following we show that there is no strategy that can guarantee a smaller deviation of the center of gravity than the strategy described in the last theorem.

Theorem 17. *The strategy given in Theorem 16 is optimal for* $n > \mu$*, i.e., there is no strategy such that the center of gravity deviates in* $[0, \frac{1}{1+\mu})$.

Proof. Because $n > \mu$, we must use at least two stacks. Now assume that we first place k items on one stack $\mathcal{S}_0$, before we start another one. Without loss of generality, we place this first k items at position $\frac{1}{1+\mu} - \varepsilon$. We proceed to show that for any $\varepsilon > 0$, we need k to be at least $\mu+1$, to get the new center of gravity to position $> -\varepsilon$ and therefore a smaller deviation as the strategy in Theorem 16.

If we place the item ℓ_{k+1} on the right side of $\mathcal{S}_0$, the new center of gravity gets to a position larger than $\frac{1}{1+\mu} - \varepsilon$, a contradiction. Thus, it must be placed on the left of $\mathcal{S}_0$. The position of this item has to be $-\frac{\mu}{1+\mu} - \varepsilon$. This yields the new center of gravity of $(k\cdot(\frac{1}{1+\mu} - \varepsilon) - \frac{\mu}{1+\mu} - \varepsilon)/k+1$. This center of gravity must be located to the right of $-\varepsilon$. Thus, we have

$$k\cdot(\frac{1}{1+\mu} - \varepsilon) - \frac{\mu}{1+\mu} - \varepsilon + (k+1)\cdot\varepsilon > 0 \quad\Leftrightarrow\quad k-\mu > 0 \quad\Leftrightarrow\quad k > \mu$$

Because we cannot stack $\mu+1$ items, we cannot have any strategy achieving a deviation of $[0, \frac{1}{1+\mu} - \varepsilon]$. We conclude that our strategy given in Theorem 16 must be optimal. □

Corollary 18. *With the given strategy for a uniform system where each item has length ℓ, the center of gravity deviates in $[0, \frac{\ell}{1+\mu}]$, which is optimal.*

4.2 Optimally Loading Without Stacking but With Minimal Space

Assume that the height of the ship to be loaded does not allow stacking items. This makes it necessary to ensure that the space consumption of the packing is minimal. We restrict ourselves to plane placements such that each state is connected. For simplicity, we assume w.l.o.g. that $\ell_1 \geqslant \cdots \geqslant \ell_n$ holds. First we argue that $\Delta(p) \geqslant \frac{\ell_2}{4}$ holds for an arbitrary connected plane placement p of S. Subsequently we give an algorithm that realizes this lower bound.

A fundamental key for this subcase is that the center of gravity of a connected plane state is the midpoint of the induced overall interval.

Observation 19. *Let s be a plane state such that the union of the corresponding intervals is an interval $[a, b] \subset \mathbb{R}$. Then $C(s) = \frac{a+b}{2}$.*

Lemma 20. *For each plane placement p of S, we have $\Delta(p) \geqslant \frac{\ell_2}{4}$.*

Proof. Let p be an arbitrary plane placement of $S = \langle (I_1, 1), \ldots, (I_n, 1) \rangle$, let $\langle s_0, s_1, \ldots, s_n \rangle$ be the sequence of states that are induced by p, and let $i, j \in \{1, \ldots, n\}$ be such that $I_i = |\ell_1|$ and $I_j = |\ell_1|$ hold. Observation 19 implies that $|C(s_{i-1}) - C(s_i)| = \frac{\ell_1}{4} \geqslant \frac{\ell_2}{4}$ and $|C(s_{j-1}) - C(s_j)| = \frac{\ell_2}{4}$. Let m_i and m_j be the midpoints of I_i and I_j. As the intervals I_i and I_j do not overlap, we conclude that $|m_i| \geqslant \frac{\ell_2}{2}$ or $|m_j| \geqslant \frac{\ell_2}{2}$ holds. W.l.o.g. assume that $|m_i| \geqslant \frac{\ell_2}{2}$ holds. This implies that $|C(s_{i-1})| \geqslant \frac{\ell_2}{4}$ or $|C(s_i)| \geqslant \frac{\ell_2}{4}$ holds. In both cases, we obtain $\Delta(p) \geqslant \frac{\ell_2}{4}$, concluding the proof. □

Lemma 21. *We can compute a placement p of S such that $\Delta(p) \leqslant \frac{\ell_2}{4}$.*

Proof. The main idea is as follows. We remember $\ell_1 \geqslant \cdots \geqslant \ell_n$ and place the items in that order. In particular, we choose the positions of the items such that $C(s_1) := -\frac{\ell_2}{4}$ and $C(s_2) := \frac{\ell_2}{4}$. The remaining intervals are placed alternating, adjacent to the left and to the right side of the previously placed intervals.

In order to show that $C(s_i) \in [-\frac{\ell_2}{4}, \frac{\ell_2}{4}]$ holds for all $i \in \{0, \ldots, n\}$, we prove by induction that $C(s_i) \in [C(s_{i-2}), C(s_{i-1})]$ holds for all odd $i \geqslant 3$ and $C(s_i) \in [C(s_{i-1}), C(s_{i-2})]$ for all even $i \geqslant 4$. As Observation 19 implies $C(s_1) = -\frac{\ell_2}{4}$ and $C(s_2) = \frac{\ell_2}{4}$, this concludes the proof.

Let $i \geqslant 3$ be odd. We have $|C(s_{i-2}) - C(s_{i-1})| = \frac{\ell_{i-1}}{2}$. This is lower bounded by $\frac{\ell_i}{2}$ because $\ell_i \leqslant \ell_{i-1}$. Furthermore, we know that $|C(s_{i-1}) - C(s_i)| = \frac{\ell_i}{2}$. This implies $C(s_i) \in [C(s_{i-2}), C(s_{i-1})]$. The argument for the case of even $i \geqslant 4$ is analogous. □

The combination of Lemma 20 and Lemma 21 implies that our approach for connected placements is optimal.

Corollary 22. *Given an arbitrary system, there is a polynomial-time algorithm for optimally loading a general set of items without stacking and under the constraint of minimal space consumption for all intermediate stages.*

4.3 Optimally Loading Exponentially Growing Items

Similar to the previous section, we consider plane placements. Now we consider the case in which the items have exponentially rising lengths. This case highlights the challenges of uneven lengths, in particular when the sizes are growing very rapidly; without special care, this can easily lead to strong deviation during the loading process. We show how the deviation can be minimized.

Theorem 23. *There is a polynomial-time algorithm for optimally loading a set of items with lengths growing exponentially by a factor $x \geqslant 2$ in increasing order w.r.t. to their lengths.*

In the following, we describe a proof of Theorem 23. In particular, we consider a system $S = \{\ell_1, \ldots, \ell_n\}$ for $n \geqslant 4$, i.e., there is an $x \geqslant 2$ such that $\ell_{i+1} = x\ell_i$ for all $i \in \{1, \ldots, n-1\}$.

First we describe the general approach of the proof and then give the details of the single steps (Lemma 25, Lemma 26, and Lemma 27) of the general approach in Sections 4.4, 4.5, and 4.6

We establish a lower bound τ for the deviation of any placement of S. The high-level idea of our approach is to place the largest interval first and slightly shifted beside 0, such that the deviation that is caused by the second largest interval is balanced; see Figure 2(a)+(b) for an illustration. Before placing the second largest interval, we place all remaining intervals in increasing order, alternating to the right side and the left side of the largest interval, such that the centers of the states alternate between $-\tau$ and τ.

In order to make the largest and second largest interval balance each other, we guarantee $C(s_1) = -\tau$ and $C(s_n) = \tau$ by considering $C(s_1) = -\tau$ and $C(s_n)$ for even n, and placing the second and the third interval on the same side such that $-\tau = C(s_1) < C(s_2) < C(s_3) = \tau$ holds for odd n.

Definition 24. $\tau := \tau(S) := \frac{\ell' + \ell''}{4\sum_{j=1}^{n} \ell_j} \ell''$.

Further proof steps are according to the following sequence of lemmas.

Lemma 25. *We have $\Delta(p) \geqslant \tau$ for each placement $p = \langle (I_1, h_1), \ldots, (I_n, h_n) \rangle$ of S.*

The following lemma guarantees that the deviation of p is equal to τ.

Lemma 26. *We have $\Delta(p) = \tau$ for the placement p computed by the above algorithm.*

Finally, we prove by Lemma 27 that the intervals in p do not overlap.

Lemma 27. *The intervals as computed by the algorithm from above are pairwise disjoint.*

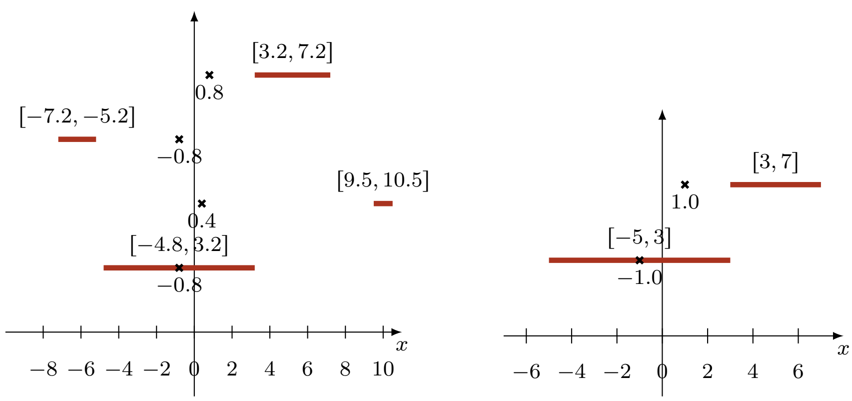
Proof of Theorem 23. The combination of the Lemmas 25, 26, and 27 guarantees that the above algorithm computes a placement with optimal deviation.

The runtime is dominated by the time needed to compute the order of $\ell_1, \ldots, \ell_n$, which takes time $\mathcal{O}(n \log n)$. $\square$

4.4 Proof of Lemma 25

Lemma 25. *We have $\Delta(p) \geqslant \tau$ for a placement $p = \langle (I_1, h_1), \ldots, (I_n, h_n) \rangle$ of S.*

In Figure 2(a) we illustrate an optimal placement p' for a 4-system S' and in Figure 2(b) an placement p'' of a 2-system S'' with $S' = \{\ell_1, \ell_2\} \subset S'' = \{\ell_1, \ell_2, \ell_3, \ell_4\}$.



(a) An optimal placement of a 4-system $S' = \{1, 2, 4, 8\}$.

(b) An optimal placement of a 2-system $S'' = \{4, 8\}$.

Figure 2: Additional intervals may improve the variation of a small placement by involving gaps between the placed intervals.

Let m_i, m'_i, and m''_i be the midpoints of the placement illustrated in Figure 2. It is important to observe that although $S'' \supset S'$, we have $m_1 \neq m'_1$ and $m_2 \neq m'_2$. In particular, the usage of the additional (smaller) intervals allow a different placement p'' that has a smaller deviation than p'. The high-level reason for this improvement is that the intervals corresponding to ℓ_3 and ℓ_4 are placed before

ℓ_2 and thus reduce the influence of ℓ_2 to the deviation of the placement illustrated in Figure 2(b). In particular, the deviation of the placements illustrated in Figure 2 determined by sum of the lengths of the two largest intervals. Furthermore, the deviation is decreased by the sum of the lengths of all intervals that are considered. Thus we chose τ as follows. Let ℓ' and ℓ'' be the largest and second largest lengths of a heterogenous system $S = \{\ell_1, \ldots, \ell_n\}$. Furthermore, let $\ell = \min\{\ell_1, \ldots, \ell_n\}$. By applying that the lengths of $\ell_1, \ldots, \ell_n$ increase constantly by a factor of $x \geqslant 2$, we make the following observation.

Observation 28. $\tau = \frac{\ell x^{2n-4}(x^2-1)}{4\sum_{i=1}^{n} \ell_i}$.

W.l.o.g., we assume $m_1 \leqslant 0$. The argument for the case $m_1 \geqslant 0$ is symmetric. Furthermore, w.l.o.g., we assume $\ell_1 = |I_1|, \ldots, \ell_n = |I_n|$ such that $\langle \ell_1, \ldots, \ell_n \rangle$.

In the following we show that $\Delta(p) \geqslant \tau$ holds if $x \geqslant 2$. To this end, we first prove that the largest interval has to be placed first in an optimal placement, see Lemma 29. Based on that, we establish that τ is a lower bound for the deviation of heterogenous systems, see Lemma 25.

As S is heterogenous, there is a unique largest length $\ell_i \in \{\ell_1, \ldots, \ell_n\}$ such that $\ell_i = \ell x^{n-1}$.

The following lemma verifies that an optimal placement p places the largest interval ℓ_i first, if τ is a tight lower bound.

Lemma 29. *If the longest block ℓ_i is not placed first, we have $\Delta(p) > \tau$.*

Proof. Note that $\ell_1, \ldots, \ell_n$ is the order that corresponds to the placement p. Suppose ℓ_i is not placed first, i.e., that $\ell_i \neq \ell_1$ holds. Based on that, we show that the center $C(s_i)$ of the state s_i is larger than τ if $x > \frac{1+\sqrt{5}}{2}$ holds. As we are considering a heterogenous system, we have $x \geqslant 2 > \frac{1+\sqrt{5}}{2}$, concluding the proof.

By applying Lemma 31 we can reformulate the statement to be shown, i.e. $|C(s_i)| > \tau$, as follows.

$$\left| \frac{C(s_{i-1}) \sum_{j=1}^{i-1} \ell_j + m_i \ell_i}{\sum_{j=1}^{i} \ell_j} \right| > \tau. \tag{1}$$

In order to show that Inequality 1 holds, we distinguish between two cases: (1) $m_i < m_1$ and (2) $m_i > m_1$. In both cases, an application of Lemma 29 concludes the proof as follows.

- $m_i < m_1$: Lemma 30 implies $m_i \geqslant -\tau + \frac{\ell_1 + \ell_i}{2} > -\tau + \frac{\ell_i}{2}$. By definition of τ we obtain $\tau < \frac{\ell_i}{2}$. Thus, we have $m_i > -\tau + \frac{\ell_i}{2} > 0$, which implies that the summand $m_i \ell_i$ in Equation 1 is positive. W.l.o.g., we assume $m(i) = -\tau + \frac{\ell_i}{2}$, $i = n$, and $C(s_{i-1}) = -\tau$, because this does not increase the left side of Inequality 1. Hence, we obtain the following.

$$\begin{aligned}
\frac{-\tau \sum_{j=1}^{n-1} \ell_j \left(-\tau + \frac{\ell_n}{2}\right)}{\sum_{j=1}^{n} \ell_j} &> \tau \\
\Leftrightarrow -\tau \sum_{j=1}^{n} \ell_j + \frac{\ell_n^2}{2} &> \tau \sum_{j=1}^{n} \ell_j \\
\Leftrightarrow \frac{\ell_n^2}{2} &> 2\tau \sum_{j=1}^{n} \ell_j
\end{aligned}$$

By applying Observation 28 and $\ell_n = \ell x^{n-1}$, we reformulate this as follows:

$$\begin{aligned}
\Leftrightarrow \frac{\left(\ell x^{n-1}\right)^2}{2} &> \frac{\ell x^{n-2}\left(\ell x^{n-1} + \ell x^{n-2}\right)}{2} \\
\Leftrightarrow x^{2n-4}\left(x^2 - x - 1\right) &> 0 \\
\Leftrightarrow x &> \frac{1+\sqrt{5}}{2}.
\end{aligned}$$

- $m_i > m_1$: This case is analogous to the previous case.

□

In the proof of Lemma 29, we apply the following auxiliary lemmas: Lemma 30, Lemma 31, and Lemma 32.

Lemma 30. *If $m_i > m_1$, we have $m_i \geqslant -\Delta(p) + \frac{\ell_1+\ell_i}{2}$. Otherwise, we have $m_i \leqslant -\frac{\ell_1+\ell_i}{2}$.*

Proof. Suppose $m_i > m_1$. As the intervals are pairwise disjoint, we have $m_i - m_1 \geqslant \frac{\ell_1+\ell_i}{2}$, which is equivalent to $m_i \geqslant m_1 + \frac{\ell_1+\ell_i}{2}$. By assumption we know $m_1 \leqslant 0$. Thus, we have $-\Delta(p) \leqslant m_1 \leqslant 0$. This implies $m_i \geqslant -\Delta(p) + \frac{\ell_1+\ell_i}{2}$.

Now assume $m_i < m_1$. As the intervals are pairwise disjoint, we have $m_1 - m_i \geqslant \frac{\ell_1+\ell_i}{2}$, which is equivalent to $m_i - m_1 \leqslant -\frac{\ell_1+\ell_i}{2}$. This implies $m_i \leqslant -\frac{\ell_1+\ell_i}{2}$, because $-m_1 \geqslant 0$.

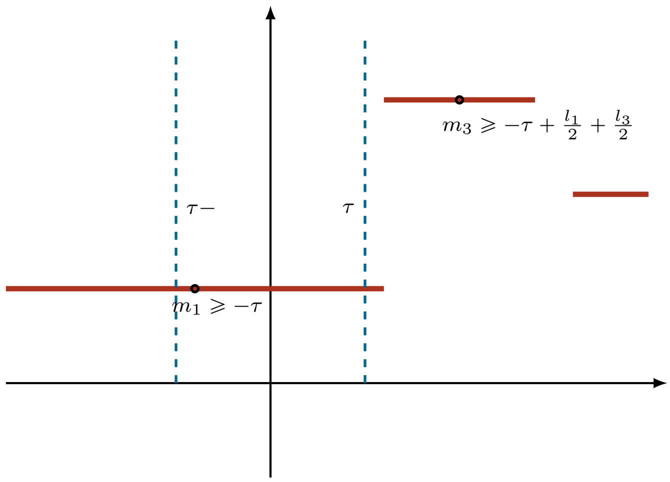


Figure 3: Estimation of the position of an interval I_3 that is not placed first and to the right of the longest interval.

□

For the two following lemmas, we consider an arbitrarily chosen but fixed $k \in \{2, \ldots, n\}$ and abbreviate $r := s_{k-1}$ and $s := s_k$.

The following lemma describes how $C(s)$ can be formulated in terms of $C(r)$.

Lemma 31. $C(s) = \frac{C(r)\left(\sum_{i=1}^{k-1} \ell_i\right) + m_k \ell_k}{\sum_{i=1}^{k} \ell_i}$.

Proof. By applying the definition of the center of gravity, we obtain the following.

$$\begin{aligned} C(s) &= \frac{\sum_{i=1}^{k} m_i \ell_i}{\sum_{i=1}^{k-1} \ell_i} \\ &= \frac{\sum_{i=1}^{k-1} m_i \ell_i + m_k \ell_k}{\sum_{i=1}^{k} \ell_i} \\ &= \frac{\left(\frac{\sum_{i=1}^{k-1} \ell_i}{\sum_{i=1}^{k-1} \ell_i}\right) \sum_{i=1}^{k-1} m_i \ell_i + m_k \ell_k}{\sum_{i=1}^{k-1} \ell_i} \\ &= \frac{C(r)\left(\sum_{i=1}^{k-1} \ell_i\right) + m_k \ell_k}{\sum_{i=1}^{k-1} \ell_i} \end{aligned}$$

□

The following lemma describes how the centers $C(r)$ and $C(s)$ uniquely determine the midpoint m_k of the k-th interval.

Lemma 32. $m_k = \frac{C(s)\sum_{i=1}^{k} \ell_i - C(r)\sum_{i=1}^{k-1} \ell_i}{\ell_k}$.

Proof. By combining the definition of $C(s)$ and Lemma 31, we obtain the following.

$$\begin{aligned}
C(s) &= \frac{\sum_{i=1}^{k} m_i \ell_i}{\sum_{i=1}^{k} \ell_i} \\
&= \frac{C(r)\sum_{i=1}^{k-1} \ell_i + m_k \ell_k}{\sum_{i=1}^{k} \ell_i} \\
\Leftrightarrow C(s)\sum_{i=1}^{k} \ell_i &= C(r)\sum_{i=1}^{k-1} \ell_i + m_k \ell_k \\
\Leftrightarrow m_k &= \frac{C(s)\sum_{i=1}^{k} \ell_i - C(r)\sum_{i=1}^{k-1} \ell_i}{\ell_k}
\end{aligned}$$

□

Lemma 33. *If $\ell_1 = \ell x^{n-1}$, $\ell_k = \ell x^{n-2}$, $m_1 < m_k$, and $\Delta(p) \leqslant \frac{\ell_k}{4}$. Then we have $m_k \geqslant C(s_{k-1}) + \frac{\ell_1+\ell_k}{2}$.*

Proof. Suppose $m_k < C(s_{k-1}) + \frac{\ell_1+\ell_k}{2}$. Observation 19 implies $|m_1| = |C(s_1)| \leqslant \Delta(p) \leqslant \frac{\ell_k}{4}$. This implies $-\frac{\ell_k}{4} \leqslant m_1 \leqslant 0$ because $m_1 \leqslant 0$. As $m_k > m_1$, we have $m_k \geqslant m_1 + \frac{\ell_1+\ell_k}{2}$. This implies $m_k \geqslant -\frac{\ell_k}{4} + \frac{\ell_1+\ell_k}{2} \geqslant -\frac{\ell_k}{4} + \frac{3\ell_k}{2} > \frac{3\ell_k}{4}$ because $\ell_1 \geqslant 2\ell_k$. Furthermore, Lemma 31 implies

$$nC(s_k) = \frac{C(s_{k-1})\sum_{j=1}^{k-1} \ell_j + m_k \ell_k}{\sum_{j=1}^{k} \ell_j}.$$

By combining the above, the lemma follows by contradiction to $\frac{\ell}{4} \geqslant \Delta(p) \geqslant |C(s_k)|$.

$$\begin{aligned}
C(s_k) &\overset{\text{Lemma 31}}{=} \frac{C(s_{k-1})\sum_{j=1}^{k-1} \ell_j + m_k \ell_k}{\sum_{j=1}^{i} \ell_j} \\
&\overset{\text{Observation 19}}{>} \frac{-\frac{\ell_k}{4}\sum_{j=1}^{k-1} \ell_j + m_k \ell_k}{\sum_{j=1}^{k} \ell_k} \\
&= \frac{-\frac{\ell_k}{4}\sum_{j=1}^{k} \ell_j}{\sum_{j=1}^{k} \ell_j} + \frac{\frac{\ell_k}{4}\ell_k}{\sum_{j=1}^{k} \ell_j} + \frac{m_k \ell_k}{\sum_{j=1}^{k} \ell_j} \\
&\overset{\sum_{j=1}^{k} \ell_k \leqslant 2\ell_k}{\geqslant} -\frac{\ell_k}{4} + \frac{\frac{\ell_k}{4}\ell_k}{\ell_k} + \frac{m_k}{2} \\
&\overset{m_k \geqslant \frac{3\ell_k}{4}}{\geqslant} \frac{\ell_k}{4}.
\end{aligned}$$

□

Based on Lemma 29, we prove that τ is a lower bound for the deviation of any placement p.
Now we are ready to prove Lemma 25.

Proof. W.l.o.g., we assume that the largest block is placed first, i.e. $\ell_i = \ell x^{n-1}$. Otherwise, Lemma 29 implies $\Delta(p) \geqslant \tau$.

Let $k \in \{1, \ldots, n\}$ be the index such that $\ell_k = \ell x^{n-2}$ is the second largest interval. Recall that $m(1) \leqslant 0$ and distinguish two cases: $m_k > m_i$ and $m_k < m_i$.

- $m_k > m_1$: The definition of $C(s_{k-1})$ implies $\sum_{j=1}^{k-1} m(j)\ell_j = C(s_{k-1})\sum_{j=1}^{k-1}\ell_j$ $(\star)$. By combining $(\star)$, the definition of the center of gravity, and Lemma 33 we can show the lemma as follows:

$$\begin{aligned}
C(m_k) &= \frac{\sum_{j=1}^{k} m_j\ell_j}{\sum_{j=1}^{k}\ell_j} \\
\Leftrightarrow \sum_{j=1}^{k-1} m_j\ell_j + m_k\ell_k &= C(s_k)\sum_{j=1}^{k}\ell_j \\
\overset{(\star)}{\Rightarrow} C(m_{k-1})\sum_{j=1}^{k-1}\ell_j + m_k\ell_k &= C(s_k)\sum_{j=1}^{k}\ell_j
\end{aligned}$$

By applying Lemma 33 we get the following.

$$\begin{aligned}
& C(s_{k-1})\sum_{j=1}^{k-1}\ell_k \\
& + \left(C(s_{k-1}) + \frac{\ell_1+\ell_k}{}\right)\ell_k \leqslant C(s_k)\sum_{j=1}^{k}\ell_j \\
\Leftrightarrow \quad & C(s_{k-1})\sum_{j=1}^{k}\ell_j + \frac{\ell_1+\ell_k}{2}\ell_k \leqslant C(s_k)\sum_{j=1}^{k}\ell_k \\
\Rightarrow \quad & \frac{\ell_1+\ell_k}{2}\ell_k \leqslant 2\max\{C(s_{k-1}), C(s_k)\}\sum_{j=1}^{k}\ell_j \\
\Leftrightarrow \quad & \frac{\ell_1+\ell_k}{4\sum_{j=1}^{k}\ell_j}\ell_k \leqslant \max\{C(s_{k-1}), C(s_k)\} \\
\overset{\substack{\ell'=\ell_1,\\ \ell''=\ell_k}}{\Rightarrow} \quad & \frac{\ell'+\ell''}{4\sum_{j=1}^{k}\ell_j}\ell_k \leqslant \Delta(p).
\end{aligned}$$

- $m_k < m_1$: Combining $m_1 \leqslant 0$, m_k, and $|m_1 m_k| \geqslant \frac{\ell_1}{\ell_k}$ leads to $m_k < -\frac{\ell_1+\ell_k}{2} < -\frac{\ell_k}{4}$. This implies $\Delta(p) > \frac{\ell_k}{4} = \frac{\ell_1+\ell_k}{4(\ell_1+\ell_k)}\ell_k \geqslant \frac{\ell_1+\ell_k}{\sum_{j=1}^{k}\ell_j}\ell_k = \tau$ as claimed.

□

4.5 Proof of Lemma 26

Lemma 26. *We have $\Delta(p) = \tau$ for the placement p computed by the algorithm described in Section 4.3.*

Proof. We have $C(s_0) = 0$. Furthermore, by Observation 19, we obtain $C(s_1) = -\tau$. In the following we distinguish between two cases: (1) n is even and (2) n is odd.

- (1) n is even: Lemma 32 implies $m_2 = \frac{C(s_2)(\ell_1+\ell_2)-C(s_1)\ell_1}{\ell_2}$. Combining this with $C(s_1) = -\tau$ implies $C(s_2) = \tau$ as follows:

$$\begin{aligned}
m_2 &= \frac{C(s_2)(\ell_1+\ell_2) - C(s_1)\ell_1}{\ell_2} \\
\Rightarrow \quad (-1)^2\left(\frac{2\tau\ell_1}{\ell_2} + \tau\right) &= \frac{C(s_2)(\ell_1+\ell_2) - C(s_1)\ell_1}{\ell_2} \\
\overset{C(s_1)=\tau}{\Leftrightarrow} \frac{\tau\ell_1 + \tau(\ell_1+\ell_2)}{\ell_2} &= \frac{C(s_2)(\ell_1+\ell_2) + \tau\ell_1}{\ell_2} \\
\Leftrightarrow \quad \tau &= C(s_2).
\end{aligned}$$

Let $i \in \{3, \ldots, n\}$. In the following we show $C(s_i) = \tau$ if i is even and $C(s_i) = -\tau$ if i is odd. Suppose $C(s_i) = \tau$ holds for all even $j \in \{4, \ldots, i-1\}$ (†) and $C(s_i) = -\tau$ holds for all odd $j \in \{3, \ldots, i-1\}$ (‡).

We first consider the case that i is even. Lemma 32 implies that m_i is equal to

$$\frac{C(s_i)\sum_{j=1}^{i} \ell_i - C(s_{i-1})\sum_{j=1}^{i-1} \ell_j}{\ell_i}$$

The algorithm guarantees $m_i = \frac{2\tau\sum_{k=1}^{i-1}\ell_k}{\ell_i} + \tau$. Furthermore, (‡) implies $C(s_{i-1}) = -\tau$. Combining the above three equations yields $C(s_i) = \tau$ as follows:

$$\begin{aligned}
\frac{2\tau\sum_{k=1}^{i-1}\ell_k}{\ell_i} + \tau \quad &= \frac{C(s_i)\sum_{j=1}^{i}\ell_i + \tau\sum_{j=1}^{i-1}\ell_j}{\ell_i} \\
\Leftrightarrow \frac{\tau\sum_{k=1}^{i-1}\ell_k + \tau\sum_{k=1}^{i}\ell_k}{\ell_i} &= \frac{C(s_i)\sum_{j=1}^{i}\ell_i + \tau\sum_{j=1}^{i-1}\ell_j}{\ell_i} \\
\Leftrightarrow \tau \quad &= C(s_i).
\end{aligned}$$

By applying a similar approach for odd i, we also obtain $C(s_i) = -\tau$.

By induction it follows $|C(s_i)| = \tau$ for all $i \in \{3, \ldots, n\}$ if n is even and thus we have $\Delta(p) = \tau$ if n is even.

- (2) n is odd: By the definition of the algorithm we have $C(s_1) < C(s_2)$ and $C(s_2) < C(s_3)$. In the following we show $C(s_3) = \tau$. This implies $|C(s_2)| \leqslant \tau$ because $-\tau = C(s_1) < C(s_2)$. Furthermore, a similar approach as in the first case implies $C(s_i) = \tau$ if i is odd and $C(s_i) = -\tau$ if i is even. Thus we obtain $\Delta(p) \leqslant \tau$ if n is odd.

  Finally, we show $C(s_3) = \tau$. By the definition of the algorithm we know that the intervals that correspond to ℓ_2 and ℓ_3 are placed adjacently on the right side of the interval that corresponds to ℓ_1. Thus, for estimating $C(s_3)$ we are allowed to consider the two intervals ℓ_2 and ℓ_3 as one interval. Let q be the midpoint of this interval. Hence, Lemma 32 implies

  $$q = \frac{C(s_1)\ell_1 + C(s_3)(\ell_1 + \ell_2 + \ell_3)}{\ell_2 + \ell_3}.$$

  Furthermore, the algorithm guarantees

  $$q = \frac{2\tau(\ell_1 + \ell_2 + \ell_3)}{\ell_3} + \tau.$$

  Combining the two last equations with $C(s_1) = -\tau$ leads to $C(q) = \tau$ as follows.

  $$\begin{aligned}
  \frac{C(s_3)(\ell_1 + \ell_2 + \ell_3) - C(s_1)\ell_1}{\ell_2 + \ell_3} &= \frac{2\tau\ell_1}{\ell_2 + \ell_3} + \tau \\
  \Leftrightarrow \frac{C(s_3)(\ell_1 + \ell_2 + \ell_3) + \tau\ell_1}{\ell_2 + \ell_3} \quad &= \frac{\tau\ell_1 + \tau(\ell_1 + \ell_2 + \ell_3)}{\ell_2 + \ell_3} \\
  \Leftrightarrow C(s_3) \quad &= \tau.
  \end{aligned}$$

  As $C(q) = C(s_3)$, we obtain $C(s_3) = \tau$. This concludes the proof.

□

4.6 Proof of Lemma 27

Lemma 27. *The intervals as computed by the algorithm from above are pairwise disjoint.*

In the following, we give a proof for Lemma 27. In particular, let $I_1, \ldots, I_n \subset \mathbb{R}$ be the intervals of lengths $\ell_1, \ldots, \ell_n$ that are computed by the algorithm. For two intervals, I_i and I_j, we abbreviate $I_i \leqslant I_j$ if $m_i < m_j$ and $|m_i - m_j| \geqslant \frac{\ell_i + \ell_j}{2}$. In the following, we show that two intervals do not overlap, i.e., that $I_i \leqslant I_j$ or $I_j \leqslant I_i$ holds for all $i \neq j \in \{1, \ldots, n\}$. We prove this separately for odd $n \geqslant 7$ and even $n \geqslant 6$ and explicitly for $n = 4$ and $n = 5$.

Lemma 34. *Let $S = \langle \ell_1, \ldots, \ell_n \rangle$ be a heterogeneous system for an even $n \geqslant 6$ and $p = \langle I_1, \ldots, I_n \rangle$ the placement that is computed by the our algorithm. Then the intervals from p are pairwise disjoint if*

- *(S1.1): $x^{n+7} + x^{n+3} + x^5 + x^4 + x^2 + 1 \geqslant 2x^{n+5} + x^{n+2} + x^n + x^7 + x^6$,*
- *(S1.2): $x^{n+5} + x^{n+2} + x^{n+1} + x^4 + x^2 \geqslant 2x^{n+4} + x^n + x^5 + x^1$, and*
- *(S1.3): $x^{n+5} + x^{n+1} + x^3 + 2x^2 + 1 \geqslant 2x^{n+3} + x^{n+2} + x^n + x^5 + x^4$.*

Proof. In the following we show that $I_3 < I_5 < \cdots < I_{n-1} < I_1 < I_n < I_{n-2} < I_{n-4} < \cdots < I_2$ holds. In order to do this we prove three implications.

- (S1.1) implies $I_3 \leqslant I_5 \leqslant \cdots \leqslant I_{n-1}$,
- (S1.2) implies $I_{n-1} \leqslant I_1$, and
- (S1.3) implies $I_n \leqslant I_{n-2} \leqslant I_{n-4} \leqslant \cdots \leqslant I_2$.

Furthermore, the inequality $I_1 \leqslant I_n$ is correct by the definition of τ. This concludes the proof.

- (S1.1) implies $I_3 \leqslant I_5 \leqslant \cdots \leqslant I_{n-1}$: Let $i \in \{3, 5, \ldots, n-3\}$ be chosen arbitrarily. We have $I_i \leqslant I_{i+2}$ if $m_i + \frac{\ell_i}{2} \leqslant m_{i+2} - \frac{\ell_{i+2}}{2}$ holds. Furthermore, above we already showed $C(s_{i-1}) = C(s_{i+1}) = \tau$ and $C(s_i) = C(s_{i+2}) = -\tau$. The algorithm guarantees

$$m_i = (-1)^i \left(\frac{2\tau \sum_{j=1}^{i-1} \ell_j}{\ell_i} + \tau \right)$$

and

$$m_{i+2} = (-1)^{i+2} \left(\frac{2\tau \sum_{j=1}^{i+1} \ell_j}{\ell_{i+2}} + \tau \right).$$

Thus, we formulate (S1.1) as a sufficient condition for $m_i + \frac{\ell_i}{2} \leqslant m_{i+2} - \frac{\ell_{i+2}}{2}$ as follows by applying $\ell_{i+2} = x^2\ell_i$ ($\star$) and the geometric sum ($\dagger$).

$$\begin{aligned}
& m_i + \frac{\ell_i}{2} && \leqslant m_{i+2} - \frac{\ell_{i+2}}{2} \\
\Leftrightarrow\ & \frac{-2\tau\sum_{j=1}^{i-1}\ell_j}{\ell_i} - \tau + \frac{\ell_i}{2} && \leqslant \frac{-2\tau\sum_{j=1}^{i+1}\ell_j}{\ell_{i+2}} - \tau - \frac{\ell_{i+2}}{2} \\
\Leftrightarrow\ & 2\tau\left(\frac{\sum_{j=1}^{i+1}\ell_j}{\ell_{i+2}} - \frac{\sum_{j=1}^{i-1}\ell_j}{\ell_i}\right) && \leqslant -\frac{1}{2}\left(\ell_{i+2} + \ell_i\right) \\
\overset{(\star)}{\Leftrightarrow}\ & 2\tau\left(\frac{1}{x^2}\sum_{j=1}^{i+1}\ell_j - \sum_{j=1}^{i-1}\ell_j\right) && \leqslant -\frac{1}{2}\left(x^2\ell_i + \ell_i\right)\ell_i \\
\Leftrightarrow\ & 2\tau\left(\frac{1}{x^2}\ell_1 - \ell_1 + \frac{1}{x^2}\sum_{j=2}^{i+1}\ell_j - \sum_{j=2}^{i-1}\ell_j\right) && \leqslant -\frac{x^2+1}{2}\ell_i^2 \\
\overset{(\dagger)}{\Leftrightarrow}\ & 2\tau\begin{pmatrix} \frac{\ell}{x^2}x^{n-1} - \ell x^{n-1} \\ +\frac{\ell}{x^2}\left(\frac{1-x^i}{1-x}\right) - \ell\left(\frac{1-x^{i-2}}{1-x}\right)\end{pmatrix} && \leqslant -\frac{x^2+1}{2}\ell_i^2 \\
\Leftrightarrow\ & 2\ell\tau\begin{pmatrix} x^{n-3} - x^{n-1} \\ +\frac{x^{-2}x^{i-2}}{1-x} - \frac{1-x^{i-2}}{1-x}\end{pmatrix} && \leqslant -\frac{x^2+1}{2}\ell_i^2 \\
\Leftrightarrow\ & 2\ell\tau\left(x^{n-3} - x^{n-1} + \frac{x^{-2}-1}{1-x}\right) && \leqslant -\frac{x^2+1}{2}\ell_i^2.
\end{aligned}$$

As ℓ_i is minimized for $i = n-3$, we substitute ℓ_i by $l_{n-3} = \ell x^{n-5}$. Furthermore, we substitute $\tau = \frac{\ell x^{2n-4}(x^2-1)}{4(x^n-1)}$. Hence

$$\begin{aligned}
& 2\ell\tau\left(x^{n-3} - x^{n-1} + \frac{x^{-2}-1}{1-x}\right) && \leqslant -\frac{x^2+1}{2}\ell_i^2. \\
\Leftrightarrow\ & 2\ell\left(\frac{\ell x^{2n-4}(x^2-1)}{4(x^n-1)}\right)\left(x^{n-3} - x^{n-1} + \frac{x^{-2}-1}{1-x}\right) && \leqslant -\left(x^2+1\right)x^{2n-10} \\
\Leftrightarrow\ & x^6\left(x^2-1\right)\left(x^{n-3} - x^{n-1} + \frac{x^{-2}-1}{1-x}\right) && \leqslant -(x^2+1)(x^n-1) \\
\Leftrightarrow\ & x^6(x^2-1)(x^{n-3}-x^{n-1}) + x^6(x^2-1)\left(\frac{x^{-1}-1}{1-x}\right) && \leqslant -(x^2+1)(x^n-1) \\
\Leftrightarrow\ & x^6(x^2-1)(x^{n-3}-x^{n-1}) - x^6(x+1)\left(x^{-2}-\right) && \leqslant -(x^2+1)(x^n-1) \\
\Leftrightarrow\ & x^{n+7} + x^{n+3} + x^5 + x^4 + x^2 + 1 && \geqslant 2x^{n+5} + x^{n+2} \\
& && \quad + x^n + x^7 + x^6.
\end{aligned}$$

- (S1.2) implies $I_{n-1} \leqslant I_1$: (S1.2) can formulated as a sufficient condition for $I_{n-1} \leqslant I_1$ as follows: $I_{n-1} \leqslant I_1$ is equivalent to $m_{n-1} + \frac{\ell_{n-1}}{2} \leqslant m_1 - \frac{\ell_2}{2}$ which can be reformulated as follows by applying

$\tau = \frac{\ell x^{2n-4}(x^2-1)}{4(x^n-1)}$ ($\star$) and the geometric sum (†).

$$\begin{aligned}
& m_{n-1} + \frac{\ell_{n-1}}{2} && \leqslant m_1 - \frac{\ell_2}{2} \\
\Leftrightarrow\ & \frac{-2\tau \sum_{j=1}^{n-2} \ell_j}{\ell_{n-1}} - \tau + \frac{\ell_{n-1}}{2} && \leqslant -\tau - \frac{\ell_1}{2} \\
\overset{(\dagger)}{\Leftrightarrow}\ & -2\tau \left(\ell \frac{1-x^{n-3}}{1-x} + \ell x^{n-1} \right) && \leqslant -\frac{\ell_{n-1}}{2}(\ell_1 + \ell_{n-1}) \\
\Leftrightarrow\ & -2\tau \left(\frac{1-x^{n-3}}{1-x} + x^{n-1} \right) && \leqslant -\frac{\ell x^{n-3}}{2} \left(\ell x^{n-1} + \ell x^{n-3} \right) \\
\overset{(\star)}{\Leftrightarrow}\ & -2\ell \left(\frac{\ell x^{2n-4}(x^2-1)}{4(x^n-1)} \right) \left(\frac{1-x^{n-3}}{1-x} + x^{n-1} \right) && \leqslant \frac{-\ell^2 x^{2n-4} - \ell^2 x^{2n-6}}{2} \\
\Leftrightarrow\ & -\left(\frac{x^{2n-4}(x^2-1)}{x^n-1} \right) \left(\frac{1-x^{n-3}}{1-x} + x^{n-1} \right) && \leqslant -x^{2n-4} - x^{2n-6} \\
\Leftrightarrow\ & \left(-x^{2n-2} + x^{2n-4} \right) \left(\frac{1-x^{n-3}}{1-x} + x^{n-1} \right) && \leqslant \begin{pmatrix} -x^{2n-4} \\ -x^{2n-6} \end{pmatrix} (x^n - 1) \\
\Leftrightarrow\ & x^{2n-7}(-x^5 + x^3) \left(\frac{1-x^{n-3}}{1-x} + x^{n-1} \right) && \leqslant x^{2n-7} \left(-x^3 - x \right) (x^n - 1) \\
\Leftrightarrow\ & \left(-x^5 + x^3 \right) \left(1 - x^{n-3} + x^{n-1} - x^n \right) && \geqslant \begin{pmatrix} -x^{n+3} + x^3 \\ -x^{n+1} + x \end{pmatrix} (1-x) \\
\Leftrightarrow\ & x^{n+5} + x^{n+2} + x^{n+1} + x^4 + x^2 && \geqslant 2x^{n+4} + x^n + x^5 + x.
\end{aligned}$$

- (S1.3) implies $I_n \leqslant I_{n-2} \leqslant I_{n-4} \leqslant \cdots \leqslant I_2$: The proof for this statement is similar to the proof of the first statement. Let $i \in \{2, 4, \ldots, n-2\}$ be chosen arbitrarily. $I_n \leqslant I_{n-2} \leqslant I_{n-4} \leqslant \cdots \leqslant I_2$ is equivalent to $m_i - \frac{\ell_i}{2} \geqslant m_{i+2} + \frac{\ell_{i+2}}{2}$. We formulate (S1.3) as a sufficient condition for $m_i - \frac{\ell_i}{2} \geqslant m_{i+2} + \frac{\ell_{i+2}}{2}$ as follows: We have $C(s_{i-1}) = C(s_{i+1}) = -\tau$ and $C(s_i) = C(s_{i+2}) = \tau$. Thus

$$\begin{aligned}
& m_i - \frac{\ell_i}{2} && \geqslant m_{i+2} + \frac{\ell_i + 2}{2} \\
\Leftrightarrow\ & \frac{2\tau \sum_{j=1}^{i-1} \ell_j}{\ell_i} + \tau - \frac{\ell_i}{2} && \geqslant \frac{2\tau \sum_{j=1}^{i+1} \ell_j}{\ell_{i+1}} + \tau + \frac{\ell_{i+2}}{2} \\
\Leftrightarrow\ & 2\tau \left(\frac{\sum_{j=1}^{i-1} \ell_j}{\ell_i} - \frac{\sum_{j=1}^{i+1} \ell_j}{\ell_{i+2}} \right) && \geqslant \frac{1}{2}(\ell_{i+2} + \ell_i) \\
\Leftrightarrow\ & 2\tau \left(\frac{\sum_{j=1}^{i-1} \ell_j}{\ell_i} - \frac{\sum_{j=1}^{i+1} \ell_j}{x^2 \ell_i} \right) && \geqslant \frac{1}{2}(x^2 \ell_i + \ell_i) \\
\Leftrightarrow\ & 2\tau \left(\ell_1 - \frac{1}{x^2}\ell_1 + \sum_{j=1}^{i-1} \ell_j - \frac{1}{x^2} \sum_{j=2}^{i+1} \ell_j \right) && \geqslant \frac{x^2+1}{2} \ell_i^2 \\
\Leftrightarrow\ & 2\tau \left(\ell_1 - \frac{1}{x^2}\ell_1 + \sum_{j=2}^{i-1} \ell_j - \frac{1}{x^2} \sum_{j=2}^{i+1} \ell_j \right) && \geqslant \frac{x^2+1}{2} \ell_i^2 \\
\Leftrightarrow\ & 2\tau \left(\begin{matrix} \ell x^{n-1} - \frac{\ell}{x^2} x^{n-1} + \\ \ell \left(\frac{1-x^{i-2}}{1-x} \right) - \frac{\ell}{x^2} \left(\frac{1-x^i}{1-x} \right) \end{matrix} \right) && \geqslant \frac{x^2+1}{2} \ell_i^2 \\
\Leftrightarrow\ & 2\ell\tau \left(x^{n-1} - x^{n-3} + \frac{1-x^{i-2}}{1-x} - \frac{x^{-2} - x^{i-2}}{1-x} \right) && \geqslant \frac{x^2+1}{2} \ell_i^2 \\
\Leftrightarrow\ & 2\ell\tau \left(x^{n-1} - x^{n-3} + \frac{1-x^{-2}}{1-x} \right) && \geqslant \frac{x^2+1}{2} \ell_i^2.
\end{aligned}$$

As ℓ_i is minimized for $i = n-2$, we substitute ℓ_i by $\ell_{n-2=\ell x^{n-4}}$. Furthermore, we substitute $\tau = \frac{\ell x^{2n-4}(x^2-1)}{4(x^n-1)}$. Hence

$$\begin{aligned}
& 2\ell\left(\frac{\ell x^{2n-4}(x^2-1)}{4(x^n-1)}\right)\left(x^{n-1}-x^{n-3}+\frac{1-x^{-2}}{1-x}\right) && \geqslant \frac{x^2+1}{2}\left(\ell x^{n-4}\right)^2 \\
\Leftrightarrow\ & \frac{x^{2n-4}(x^2-1)}{x-1}\left(x^{n-1}-x^{n-3}+\frac{1-x^{-2}}{1-x}\right) && \geqslant (x^2+1)(x^{2n-8}) \\
\Leftrightarrow\ & x^4(x^2-1)\left(x^{n-1}-x^{n-3}+\frac{1-x^{-2}}{1-x}\right) && \geqslant (x^2+1)(x^n-1) \\
\Leftrightarrow\ & x^4(x^2-1)(x^{n-1}-x^{n-3})+x^4(x^2-1)\left(\frac{1-x^{-2}}{1-x}\right) && \geqslant (x^2+1)(x^n-1) \\
\Leftrightarrow\ & (x^6-x^4)(x^{n-1}-x^{n-3})-x^4(x+1)(1-x^{-2}) && \geqslant (x^2+1)(x^n-1) \\
\Leftrightarrow\ & x^{n+5}+x^{n+1}+x^3+2x^2+1 && \geqslant 2x^{n+3}+x^{n+2} \\
& && \quad +x^n+x^5+x^4.
\end{aligned}$$

□

Lemma 35. *Let $S = \langle \ell_1, \dots, \ell_n \rangle$ be a heterogeneous system for $n = 4$ and $p = \langle I_1, \dots, I_n \rangle$ the placement that is computed by the algorithm. Then the intervals from p are pairwise disjoint if*

- *(S1.2): $x^{n+5} + x^{n+2} + x^{n+1} + x^4 + x^2 \geqslant 2x^{n+4} + x^n + x^5 + x^1$, and*
- *(S1.3): $x^{n+5} + x^{n+1} + x^3 + 2x^2 + 1 \geqslant 2x^{n+3} + x^{n+2} + x^n + x^5 + x^4$.*

Proof. Similar to the proof of Lemma 34, we guarantee $I_3 \leqslant I_5 \leqslant \cdots \leqslant I_{n-1} \leqslant I_1 \leqslant I_n \leqslant I_{n-2} \leqslant \cdots \leqslant I_2$. As $n = 4$, we do not have to take care about $I_3 \leqslant \cdots \leqslant I_{n-3} \leqslant I_{n-1}$. The proof for (S1.2) and (S1.3) is the same as in the proof of Lemma 34. □

Lemma 36. *Let $S = \langle \ell_1, \dots, \ell_n \rangle$ be a heterogeneous system for an odd $n \geqslant 7$ and $p = \langle I_1, \dots, I_n \rangle$ the placement that is computed by the algorithm. Then the intervals from p are pairwise disjoint if*

- *(S1.1): $x^{n+7} + x^{n+3} + x^5 + x^4 + x^2 + 1 \geqslant 2x^{n+5} + x^{n+2} + x^n + x^7 + x^6$,*
- *(S1.2): $x^{n+5} + x^{n+2} + x^{n+1} + x^4 + x^2 \geqslant 2x^{n+4} + x^n + x^5 + x^1$,*
- *(S1.3): $x^{n+5} + x^{n+1} + x^3 + 2x^2 + 1 \geqslant 2x^{n+3} + x^{n+2} + x^n + x^5 + x^4$, and*
- *(S1.4): $x^{2n}(x^{-2} - x^{-4})(x^{n+2} - x^n - x^{n-1} - x^3 - 2x^2 - 2x - 1) \geqslant (x^3 + x + 1)(x^n - 1)(x^3 + x^4)$.*

Proof. (S1.1), (S1.2), and (S1.3) are the same conditions as in Lemma 34. By considering the second and third interval as one interval, we are allowed to apply an approach that is similar to the argument from the proof of Lemma 34. In particluar, except for the first three intervals we still have the property that lengths of the intervals increase by a factor of $x \geqslant 2$.

We show that $I_3 \leqslant I_5 \leqslant \cdots \leqslant I_{n-4} \leqslant I_{n-2} \leqslant I_1 \leqslant I_{n-1} \leqslant I_{n-3} \leqslant I_{n-5} \leqslant \cdots \leqslant I_4 \leqslant I_2$ holds if (S1.1), (S1.2), (S1.3), and (S1.4) are fulfilled. By the above argument, the lengths of the intervals are $\ell x^{n-2}, \ell + \ell x, \ell x^2, \ell x^3, \dots, \ell x^{n-2}$, which means that we are now considering $n-1$ intervals.

In order to show $I_3 \leqslant I_5 \leqslant \cdots \leqslant I_{n-4} \leqslant I_{n-2} \leqslant I_1 \leqslant I_{n-1} \leqslant I_{n-3} \leqslant I_{n-5} \leqslant \cdots \leqslant I_4 \leqslant I_2$, we prove four implications:

- (S1.1) implies $I_3 \leqslant I_5 \leqslant \cdots \leqslant I_{n-4} \leqslant I_{n-2}$,
- (S1.2) implies $I_{n-2} \leqslant I_1$,
- (S1.3) implies $I_{n-1} \leqslant I_{n-3} \leqslant \cdots \leqslant I_4 \leqslant I_2$, and
- (S1.4) implies $I_4 \leqslant I_2$.

The inequality $I_1 \leqslant I_n$ is correct by the definition of τ. This concludes the proof.

- (S1.1) implies $I_3 \leqslant I_5 \leqslant \cdots \leqslant I_{n-4} \leqslant I_{n-2}$: The argument is the same as in the proof of Lemma 34, where we lower bound ℓ_i by $\ell_{n-2} = \ell x^{n-5}$.
- (S1.2) implies $I_{n-2} \leqslant I_1$: The argument is the same as in the proof of Lemma 34, where we substitute $n-1$ by $n-2$.
- (S1.3) implies $I_{n-1} \leqslant I_{n-3} \leqslant \cdots \leqslant I_4 \leqslant I_2$: Similar to the approach for (S1.1), the argument is the same as in the proof of Lemma 34, where we lower bound ℓ_i by $\ell_{n-2} = \ell x^{n-5}$.
- (S1.4) implies $I_4 \leqslant I_2$: $I_4 \leqslant I_2$ is equivalent to $m_2 - \frac{\ell_2}{2} \geqslant m_4 + \frac{\ell_4}{2}$. We have $C(s_1) = C(s_3) = -\tau$ and $C(s_2) = C(s_4) = \tau$. Thus:

$$
\begin{aligned}
& m_2 - \frac{\ell_2}{2} \\
& \geqslant m_4 + \frac{\ell_4}{2} \\
\Leftrightarrow \quad & \frac{2\tau\ell_1}{\ell_2} + \tau - \frac{\ell_2}{2} \\
& \geqslant \frac{2\tau(\ell_1+\ell_2+\ell_3)}{\ell_4} + \tau + \frac{\ell_4}{2} \\
\Leftrightarrow \quad & 2\tau\left(\frac{\ell_1}{\ell_2} - \frac{\ell_1+\ell_2+\ell_3}{\ell_4}\right) \\
& \geqslant \frac{1}{2}(\ell_2 + \ell_2) \\
\Leftrightarrow \quad & 2\tau\left(\frac{\ell x^{n-1}}{\ell + \ell x} - \frac{2\left(x^{n-1}+1+x+x^2\right)}{\ell x^3}\right) \\
& \geqslant \frac{1}{2}(x^3+1+x) \\
\Leftrightarrow \quad & 2\frac{\ell x^{2n-4}(x^2-1)}{4(x^n-1)}\left(\frac{x^{n-1}}{1+x} - \frac{x^{n-1}+1+x+x^2}{x^3}\right) \\
& \geqslant \frac{1}{2}\left(x^3+1+x\right) \\
\Leftrightarrow \quad & \left(x^{2n-2} - x^{2n-4}\right)\left(\frac{x^{n-1}}{1+x} - \frac{x^{n-1}+1+x+x^2}{x^3}\right) \\
& \geqslant \left(x^3+1+x\right)\left(x^n-1\right) \\
\Leftrightarrow \quad & \left(x^{2n-2} - x^{2n-4}\right)\left(\frac{x^{n+2} - x^{n-1} - 1 - 2x^2 - xx^n - 2x - x^3}{x^3+x^4}\right) \\
& \geqslant (x^3+1+x)(x^n-1) \\
\Leftrightarrow \quad & x^{2n}(x^{-2} - x^{-4})(x^{n+2} - x^n - x^{n-1} - x^3 - 2x^2 - 2x - 1) \\
& \geqslant (x^3+x+1)(x^n-1)(x^3+x^4).
\end{aligned}
$$

□

Lemma 37. *Let $S = \langle \ell_1, \ldots, \ell_n \rangle$ be a heterogeneous system for $n = 5$ and $p = \langle I_1, \ldots, I_n \rangle$ the placement that is computed by the algorithm. Then the intervals from p are pairwise disjoint if*

- *(S1.2):* $x^{n+5} + x^{n+2} + x^{n+1} + x^4 + x^2 \geqslant 2x^{n+4} + x^n + x^5 + x^1$,
- *(S1.3):* $x^{n+5} + x^{n+1} + x^3 + 2x^2 + 1 \geqslant 2x^{n+3} + x^{n+2} + x^n + x^5 + x^4$, *and*
- *(S1.4):* $x^{2n}(x^{-2} - x^{-4})(x^{n+2} - x^n - x^{n-1} - x^3 - 2x^2 - 2x - 1) \geqslant (x^3+x+1)(x^n-1)(x^3+x^4)$.

Proof. Similar to the proof of Lemma 36, we guarantee $I_3 \leqslant I_5 \leqslant \cdots \leqslant I_{n-2} \leqslant I_1 \leqslant I_{n-1} \leqslant I_{n-3} \leqslant \cdots \leqslant I_4 \leqslant I_2$. As $n = 5$, we do not have to deal with $I_3 \leqslant \cdots \leqslant I_{n-2}$. The proof for (S1.2), (S1.3), and (S1.4) is the same as in the proof of Lemma 36. □

Now we are ready to give the proof of Lemma 27:

Proof of Lemma 27. By combining Lemma 34, 35, 36, and 37 we obtain that the intervals of p are pairwise disjoint because the Inequations (S1.1), (S1.2), (S1.3), and (S1.4) are fulfilled for $x \geqslant 2$. $\square$

5 Conclusion

We have introduced a new family of problems that seek to balance objects, controlling the variation of their center of gravity during the loading and unloading of the objects. We have provided hardness results and optimal or constant-factor approximation algorithms.

There are various related challenges. These include sequencing problems with multiple loading and unloading stops (which arise in vehicle routing or tour planning for container ships); variants in which items can be shifted in a continuous fashion; batch scenarios in which multiple items are loaded or unloaded at once (making it possible to maintain better balance, but also increasing the space of possible choices); and higher-dimensional variants, possibly with inhomogeneous space constraints. All these are left for future work.

Acknowledgements.

We would like to thank anonymous reviewers of the conference abstract for providing helpful comments and suggestions improving the presentation of this paper.

References

[1] S. V. Amiouny, J. J. Bartholdi, J. H. V. Vate, and J. Zhang. Balanced loading. *Operations Research*, 40(2):238–246, 1992.

[2] E. E. Bischoff and M. D. Marriott. A comparative evaluation of heuristics for container loading. *Eur. J. Oper. Res.*, 44(2):267–276, 1990.

[3] E. E. Bischoff and M. Ratcliff. Issues in the development of approaches to container loading. *Omega*, 23(4):377–390, 1995.

[4] S. G. Christensen and D. M. Rousøe. Container loading with multi-drop constraints. *International Transactions in Operational Research*, 16(6):727–743, 2009.

[5] A. P. Davies and E. E. Bischoff. Weight distribution considerations in container loading. *Eur. J. Oper. Res.*, 114(3):509–527, 1999.

[6] G. Fasano. A MIP approach for some practical packing problems: Balancing constraints and tetris-like items. *4OR*, 2(2):161–174, 2004.

[7] H. Gehring and A. Bortfeldt. A genetic algorithm for solving the container loading problem. *Intern. Transactions in Operational Research*, 4(5-6):401–418, 1997.

[8] H. Gehring, K. Menschner, and M. Meyer. A computer-based heuristic for packing pooled shipment containers. *Eur. J. Oper. Res.*, 44(2):277–288, 1990.

[9] P. Gilmore and R. E. Gomory. Multistage cutting stock problems of two and more dimensions. *Operations research*, 13(1):94–120, 1965.

[10] S. Limbourg, M. Schyns, and G. Laporte. Automatic aircraft cargo load planning. *JORS*, 63(9):1271–1283, 2012.

[11] V. Lurkin and M. Schyns. The airline container loading problem with pickup and delivery. *Eur. J. Oper. Res.*, 244(3):955–965, 2015.

[12] K. Mathur. An integer-programming-based heuristic for the balanced loading problem. *Oper. Res. Lett.*, 22(1):19–25, 1998.

[13] M. Mongeau and C. Bes. Optimization of aircraft container loading. *IEEE Transactions on Aerospace and Electronic Systems*, 39(1):140–150, 2003.

[14] B. O. Øvstebø, L. M. Hvattum, and K. Fagerholt. Optimization of stowage plans for roro ships. *Computers & Operations Research*, 38(10):1425–1434, 2011.

[15] B. O. Øvstebø, L. M. Hvattum, and K. Fagerholt. Routing and scheduling of roro ships with stowage constraints. *Transportation Research Part C: Emerging Technologies*, 19(6):1225–1242, 2011.

[16] M. Paterson, Y. Peres, M. Thorup, P. Winkler, and U. Zwick. Maximum overhang. *The American Mathematical Monthly*, 116(9):763–787, 2009.

[17] M. Paterson and U. Zwick. Overhang. *The American Mathematical Monthly*, 116(1):19–44, 2009.

[18] H. Pollaris, K. Braekers, A. Caris, G. K. Janssens, and S. Limbourg. Vehicle routing problems with loading constraints: state-of-the-art and future directions. *OR Spectrum*, 37(2):297–330, 2015.

[19] S. Sevastianov. On some geometric methods in scheduling theory: a survey. *Discrete Applied Mathematics*, 55(1):59–82, 1994.

[20] S. Sevastianov. Nonstrict vector summationin multi-operation scheduling. *Annals of Operations Research*, 83:179–212, 1998.

[21] W. Souffriau, P. Demeester, G. V. Berghe, and P. De Causmaecker. The aircraft weight and balance problem. *Proceedings of ORBEL*, 22:44–45, 1992.

[22] W. Vancroonenburg, J. Verstichel, K. Tavernier, and G. V. Berghe. Automatic air cargo selection and weight balancing: a mixed integer programming approach. *Transp. Research Part E: Logistics and Transportation Review*, 65:70–83, 2014.