Internship report: Approximability of Max-Min Allocation under Equal Valuations

Kimon Böhmer
Supervisor: Chien-Chung Huang
DI ENS

March-August 2025

Summary

General context

ITEM ALLOCATION is a fundamental problem of combinatorial optimization. Its task consists in distributing a set of *items* among a set of *players*, each equipped with a valuation function over subsets of items, with the goal of optimizing a specific objective function. The most classical objective functions are *max-sum*, i.e. maximizing the average satisfaction across all players, and *max-min*, maximizing the satisfaction of the least satisfied player. This work focusses on the latter objective function, which is less studied. Applications of max-min allocation range from assigning classrooms to charter schools [16] over sensor placement and scheduling [15] to allocating resources to GPUs for deep learning training [8].

Naturally, if players are allowed to have arbitrary valuation functions, no useful algorithmic guarantees can be provided. As a result, research focusses on restricted classes of valuation functions that are both expressive and computationally tractable. For example, a common assumption is that each player has an *additive* valuation function; that is, each item is associated to a certain weight, and the satisfaction of the player is the sum of the weights of the received items. Another important and more general class of valuation functions are *submodular* functions; besides their useful algorithmic properties, they naturally model the law of *diminishing returns* in economics. This makes them especially interesting in the context of allocation.

In this work, we will mostly restrict our attention to the case where all players share the same valuation function. Assuming that the valuation function is submodular, prior to this internship, the state of the art was a $\frac{e-1}{3e} \approx 0.21$-approximation [15].

The report also explores the integrality gap of the so-called *configuration linear program*, an important linear programming (LP) relaxation for max-min allocation. The study of integrality gaps of natural LP relaxations is motivated by the fact that they often lead to new and better approximation algorithms, especially in the context of the configuration LP [5]. Previous work had established that for identical additive valuations, this gap is between $\frac{1023}{1022}$ and 2 [17].

Research problem

Specific questions studied included: Can we improve the approximation algorithm of [15], using combinatorial methods or mathematical programming approaches via the configuration LP?

Can we improve the bounds on the integrality gap of the configuration LP, or establish new results for related problems in max-min allocation, by applying similar techniques?

Max-min allocation is a problem which arises very naturally from economic motivations and has many real-world applications. The problem has its root at machine scheduling and is therefore an old problem in combinatorial optimization. However, its approximability remains extremely poorly understood, compared to other classical problems. This made me particularly intrigued by the specific problem.

Contribution

Our contributions can be categorized as follows:

- In Section 2, we improve the state of the art for submodular max-min allocation with identical valuations, by showing that a certain greedy algorithm gives a $\frac{2}{5} = 0.4$-approximation for submodular max-min allocation. This is the most important result of the internship. For the special case of weighted matroid rank functions, we provide a $\frac{1}{2}$-approximation.
- In Section 3, we use a new technique to prove upper bounds for the integrality gap of the configuration LP, based on *dual fitting*. With this method, we show that the integrality gap is upper bounded by 1.5, establishing that this LP is strictly stronger than the other classical LP relaxations previously studied. Additionally, we improve the lower bound on the integrality gap to $\frac{73}{72}$ and prove bounds for more general settings like submodular functions, or players with different capacities (also called *parallel machines*).
- During the internship, we also obtained several results in the setting of so-called *constrained agents*, which combines the max-sum and max-min objectives by partitioning the players into two groups. Since this topic is less related to the rest of the report, we only briefly summarize these results in Section 4, and discuss them in more detail in the appendix.

Arguments supporting its validity

We provide proofs for all of our claims. Due to space constraints, we only keep the technically most interesting proofs in the main part, and defer the other proofs to the appendix.

Summary and future work

We contributed to a better understanding of approximation algorithms for max-min allocation with identical additive or submodular valuation functions. However, several natural questions remain open. First, the proposed 0.4-approximation is still far from the best-known complexity-theoretical hardness bound of $1 - \frac{1}{e} \approx 0.63$. Closing this gap remains the most pressing open problem. We believe that a local search algorithm based on the same *max-sum to max-min* paradigm as our greedy algorithm might lead to a better ratio.

Another promising direction is to further explore the configuration LP. It would be valuable to determine the exact value of its integrality gap in the case of identical additive valuations. Our work suggests that the true gap might be much closer to the lower bound than to the current upper bound (see proposition 7). It would also be interesting to extend some of the developped proof techniques to more general settings, such as submodular functions and parallel machines. This could lead to new state-of-the-art approximation algorithms for several important problems.

0.1 Approximability landscape

After this brief summary, we give a slightly more profound study of related work and explain how our results fit into the global complexity picture. Table 1 gives an overview over lower and upper bounds for the approximability of max-min allocation, as well as for the integrality gap of the configuration LP, under different variants of the problem.

Class	**Algorithms**	**Hardness**	**Integrality Gap Upper Bound**	**Integrality Gap Lower Bound**
Additive	PTAS [22]	**NP**-hard	3/2 (Theorem 3)	73/72 (Theorem 4)
Parallel	PTAS [3]	**NP**-hard	2 (Proposition 4)	73/72
Restricted Assignment	1/2-approx (Theorem 2)	**NP**-hard	2	73/72
Submodular	2/5-approx (Theorem 1)	$1 - 1/e + \varepsilon$ [14]	?	4/3 (Proposition 6)
Restricted Assignment	$(1/4) - \varepsilon$-approx [10]	$1/2 + \varepsilon$ [6]	3.534 [13]	2

Table 1: Approximability and integrality gap of the configuration LP for different variants of max-min allocation.

Additive valuations. Additivity is one of the most simple assumptions on the valuation functions of the players. In the general version where each player might have a different additive valuation function, the best-known algorithm by [7] achieves an approximation factor of $\mathcal{O}(n^{1+\varepsilon})$. Notably, computational complexity only rules out a factor better than 2, as shown in [6].

A well-studied special case is the so-called *restricted assignment* case, where the valuation of a player for an item j is either $f(j)$ or 0. To tackle this problem, Bansal and Sviridenko [5] introduced the configuration LP, and subsequent work [5, 2, 19, 1] led to the current best approximation factor of $\frac{1}{4} - \varepsilon$ for the problem [10]. Recently, the integrality gap of the configuration LP was shown to be even smaller [13].

In the simpler setting where each player has the same additive valuation function, Woeginger designed a polynomial-time approximation scheme (PTAS) [22]. The PTAS was later generalized to the case of so-called *parallel machines* [3], where each 'machine' can have a different *speed*, or, from an allocation point of view, each player can be satisfied with a different amount of value.

Submodular valuations. Submodular valuations have also been studied. The first approximation algorithm for the general case was given in [12], and even though improvements have been made [11], the best approximation factor is still polynomial in the input. Notably, no better hardness than 2 is known [6]. In the special case where the number of players is constant, an optimal $1 - 1/e$-approximation has been found [9]. For max-sum, Vondrak gave a $1 - 1/e$-approximation [21], which is optimal under $\mathbf{P} \neq \mathbf{NP}$ [14], but even unconditionally using information-theoretical lower bounds [18]. In the restricted assignment case, [4] managed to generalize the techniques from Bansal and Sviridenko to obtain a $\mathcal{O}(\log \log n)$-approximation.

When all players have the same valuation function, [15] gave the first constant-factor approximation algorithm, which uses an algorithm for max-sum as a subroutine. Plugging in Vondrak's optimal algorithm, their method achieves a $\frac{e-1}{3e} \approx 0.21$-approximation.

Focus of this work and results. Our goal in this work is to better understand the approximability of the above-mentioned problems in the case where all players have equal, or closely related, valuation functions. This restriction has real-world motivations [15] but is also interesting since various hardness results already hold in this simple case [14, 18].

As already mentioned in the summary, we significantly improve on the result of [15] by giving a $\frac{2}{5}$-approximation for the problem of max-min allocation with equal submodular valuations. Our algorithm is essentially greedy and rather easy to describe, but the analysis is quite involved.

Furthermore, we study the integrality gap of the configuration LP with equal preferences. The other standard LP relaxation, called *assignment LP*, has an integrality gap of two, and no better integrality gap was known for the configuration LP in this setting. We prove that the configuration LP is strictly more powerful than the assignment LP, even in the case of equal valuations, by showing that the integrality gap of the configuration LP is at most 1.5. We also give other lower and upper bounds to the integrality gap in various settings, as seen in table 1.

0.2 General structure of the report

In Section 1, we give basic definitions and formally introduce the computational problem. In Section 2, we discuss greedy algorithms for Submodular Max-Min Allocation, showing the aforementioned $\frac{2}{5}$-approximation (theorem 1). We also prove theorem 2 about weighted matroid rank functions. In Section 3, we study the configuration LP and prove upper and lower bounds on its integrality gap, for example theorems 3 and 4 and proposition 6. Finally, in Section 4, we present our results in the *constrained agent setting*, in particular theorems 5 and 6.

We assume that the reader is familiar with basic notions in approximation algorithms and optimization, such as (dual) linear program, integrality gap and separation oracle. However, we give a reminder for the most basic notions in Appendix D.

1 Preliminaries

Consider a set J (with $n := |J|$) of items and a *valuation function* $f : 2^J \to \mathbb{Q}$ that assigns a value to each subset of J. We simplify notation by writing $f(j)$ instead of $f(\{j\})$, for $j \in J$.

We focus on two cases: First, when f is *additive*, that is, $f(S \cup T) = f(S) + f(T) - f(S \cap T)$. In this case, we can write $f(S) = \sum_{s \in S} f(s)$, for any $S \subseteq J$. Our second case concerns submodular functions. A function $f : 2^J \to \mathbb{Q}$ is called *submodular*, if for every two subsets S and T,

$$f(S \cup T) \leq f(S) + f(T) - f(S \cap T).$$

Closely related to submodular functions is the notion of *marginal contribution*. We define the marginal contribution on f of an item j with respect to a subset S as $\Delta_f(j \mid S) := f(S \cup \{j\}) - f(S)$. We extend this definition to sets (i.e. $\Delta_f(S \mid T) = f(S \cup T) - f(T)$). In fact, a useful characterization of submodularity is that for all subsets $S \subseteq T$ of J and $j \in J$,

$$\Delta_f(j \mid S) \geq \Delta_f(j \mid T).$$

A function is *monotone* if for all $S \subseteq T$, we have $f(S) \leq f(T)$. We assume $f(\emptyset) = 0$, and thus a monotone function is also non-negative. We also consider a set P (with $m := |P|$) of *players*. Each player p has a valuation function f_p describing his satisfaction with a certain set of items.

More formally, a *partial allocation* of J is a partition $A = (A_1, ..., A_m)$ of a subset of J. Slightly abusing notation, we write $j \in A$ for $j \in \bigcup_{i \in P} A_i$. If every item is allocated, we speak of an *allocation*. We refer to the disjoint sets of a (partial) allocation as *configurations*. We define $f_{\text{sum}}(A) = \sum_{p\in P} f_p(A_p)$, $f_{\min}(A) := \min_{p\in P} f_p(A_p)$ and $f_{\max}(A) = \max_{p\in P} f_p(A_p)$. We will refer to players p with $f_p(A_p) = f_{\min}(A)$ as *min players* (and the same for *max players*).

1.1 Submodular Max-Min Allocation

With these definitions, we are ready to define the main computational problem of interest.

SUBMODULAR MAX-MIN ALLOCATION
Input: A set of items J, a set of players $P = \{1, ..., m\}$, and a monotone submodular function $f : 2^J \to \mathbb{Q}$.
Output: An allocation A that maximizes $f_{\min}(A)$

Notice that each player has the same valuation function f. We assume that we are given access to a so-called *value oracle*, which, given any $S \subseteq J$, returns $f(S)$. [1] To see **NP**-hardness, consider the classical **NP**-hard PARTITION problem: We are given a multiset of integers and ask to partition it into two parts of equal value. This clearly corresponds to the special case of our problem where there are only two players and their valuation function is additive.

Approximation Hardness. Apart from the general **NP**-hardness, it was shown that the max-sum version of the problem is **NP**-hard to approximate within any factor better than $1 - \frac{1}{e}$ [14]. In the reduction, each player shares the same valuation function, and moreover gets the same value. Their hardness result therefore immediately translates to our setting:

Corollary 1 ([14])**.** *Unless $\boldsymbol{P} = \boldsymbol{NP}$,* SUBMODULAR MAX-MIN ALLOCATION *cannot be approximated to factor $1 - \frac{1}{e} + \varepsilon$ for any $\varepsilon > 0$.*

1.2 Facts about Submodular Functions

In this subsection, we state some useful (generally already well-known) facts about submodular functions. Let $f : 2^J \to \mathbb{Q}$ be submodular and monotone.

Observation 1. *The restriction of f to any subset $J' \subseteq J$ is submodular and monotone.*

Lemma 1. *Consider a new item z. For any $t \in \mathbb{Q}$, define the function $f_t : 2^{J\cup\{z\}} \to \mathbb{Q}$ by $f_t(S \cup \{z\}) = f(S) + t$ and $f_t(S) = f(S)$ for all $S \not\ni z$. Then, f_t is submodular and monotone.*

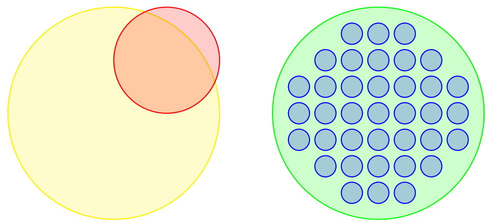
Proof. Monotonicity being clear, we only prove submodularity.

$$\begin{aligned} f_t(S) + f_t(T) &= f(S) + f(T) + (\mathbb{1}_{z\in S} + \mathbb{1}_{z\in T}) \cdot t \\ &\geq f(S \cup T) + f(S \cap T) + (\mathbb{1}_{z\in S} + \mathbb{1}_{z\in T}) \cdot t \\ &= f(S \cup T) + f(S \cap T) + (\mathbb{1}_{z\in S\cup T} + \mathbb{1}_{z\in S\cap T}) \cdot t \\ &= f_t(S \cup T) + f_t(S \cap T) \end{aligned}$$

□

[1] We note that in the literature (especially in work motivated by economic problems), a stronger type of oracle exists, namely the *demand oracle*: It is assumed that each item j carries a price p_j, and the oracle returns the set S which maximizes $f(S) - \sum_{j\in S} p_j$.

Figure 1: The player with the yellow base item will prefer the red item over any blue item, and thus the red item, which is very beneficial for the second player with the green base item, will not be available anymore.



Observation 2. $\Delta_f(S \mid T) - f(S) = f(S \cup T) - f(T) - f(S) = \Delta_f(T \mid S) - f(T)$.

Observation 3. *Let* $S = \{j_1, ..., j_k\}$. *Then,* $\Delta_f(S \mid T) = \sum_{i=1}^{k} \Delta_f(j_i \mid T \cup \{j_1, ..., j_{i-1}\})$.

A direct corollary by using submodularity is:

Observation 4. $\Delta_f(S \mid T) \leq \sum_{j \in S} \Delta_f(j \mid T)$.

Observation 5. $f(S \cup T) = f(S) + f(S \cup T) - f(S) = f(S) + \Delta_f(T \mid S)$.

2 Greedy approaches

In this section, we investigate greedy algorithms to solve Submodular Max-Min Allocation.

2.1 The Natural Greedy Algorithm

Certainly, the most natural greedy algorithm to consider always picks the current min player and assigns to them the item with the highest marginal contribution. We call this simple algorithm the *natural greedy algorithm*. Unfortunately, even though it appears to be very promising, this algorithm can produce arbitrarily bad solutions. We give here a high-level intuition, and defer the full counter-example and the proof to Appendix A.1.

Proposition 1. *The natural greedy algorithm has no performance guarantee better than* $1/m$.

Intuition. In our counterexample, each item covers a certain area on the plane, and the value of a set of items is the amount of area covered by the union of all items. Consider the example in figure 1, where one player already got the yellow item, and the other player already got the green item. Clearly, to increase its area, the yellow player should choose the blue items, while the green player should choose the red item. But, assuming the yellow player is the current min player, they will choose the single item which maximizes the current marginal contribution, that is the red item. The fundamental problem is that the items which are "good" for a certain player might have extremely small marginal contribution in comparison to other items. In other words, the good items are *many*, but every single one of them is bad.

2.2 Truncated Max-Sum Greedy

The above counterexample shows that a greedy algorithm that only considers the min player is too restrictive. We should rather allow the greedy algorithm to allocate items to non-min players. We therefore investigate the worst-case performance of the following greedy algorithm: Fix a threshold $\alpha \leq 1$. We assume that the algorithm knows the optimal value OPT, by performing a binary search. First allocate one of the m biggest items to each player. Then greedy allocates according to the following rule: Always choose the pair of available item j and player p that maximizes the marginal contribution $\Delta_f(j \mid A_p)$, under the constraint that $f(A_p) < \alpha \cdot \text{OPT}$, i.e. we only consider players that have not reached an α-fraction of their desired value.

Then, the goal is to show that each player will eventually surpass this threshold, which proves an α-approximation. The pseudocode is shown in algorithm 1.

Algorithm 1: The truncated max-sum greedy

Data: items J, players P, submodular function $f : 2^J \to \mathbb{Q}$
Result: an allocation A
Remove the m items j with largest $f(j)$ from J, and give one of them to each player;
while $J \neq \emptyset$ *and there exists* $p \in P$ *with* $f(A_p) < \alpha \cdot OPT$ **do**
  $(j^*, p^*) \leftarrow \arg\max_{j \in J, p \in P : f(A_p) < \alpha \cdot \text{OPT}} \Delta_f(j \mid A_p)$;
  $J \leftarrow J - \{j^*\}$;
  $A_{p^*} \leftarrow A_{p^*} \cup \{j^*\}$;
end
return $(A_1, ..., A_m)$

Observation 6. *If the greedy gives an α-approximation on instances where for all items $j \in J$, $f(j) < \alpha \cdot OPT$, then it also gives an α-approximation for the general case.*

Proof. Let OPT denote the optimal value in the general instance. Assume that there are k *big* items j with $f(j) \geq \alpha \cdot \text{OPT}$. In an optimal allocation, clearly these items can be given to at most k players. Thus, there exists an instance on $m - k$ players and a subset $J' \subseteq \{j \in J \mid f(j) < \alpha \cdot \text{OPT}\}$ of items that achieve at least OPT.

If greedy is performed on the instance with $m - k$ players and without the big items, then it will obtain a solution of value at least $\alpha \cdot \text{OPT}$ (since the optimal solution on this reduced instance has value OPT, as argued above). Adding k players, each containing one big item, gives a solution of value at least $\alpha \cdot \text{OPT}$ to the general instance. □

As a warm-up, we will first show that we obtain a $\frac{1}{3}$-approximation when setting $\alpha = \frac{1}{3}$. Afterwards, we will show that we can actually obtain a $\frac{2}{5}$-approximation with $\alpha = \frac{2}{5}$.

Consider a greedy allocation A and an optimal allocation A^*. Let q be the min player in A, i.e. $f(A_q) = f_{\min}(A)$. Throughout the proof, we will use the following notation: The t-th item which got allocated to player p by the greedy algorithm is denoted by a_p^t.

We first show the following simple but crucial lemma.

Lemma 2. *Assume $f(A_q) < \alpha \cdot OPT$. For all players $p \in P$,*

$$\sum_{j \in A_p} \Delta_f(j \mid A_q) \leq f(A_p)$$

Proof. Let $A^{(t)}$ denote the partial allocation of the algorithm after a_p^t has been allocated.

$$\sum_{j \in A_p} \Delta_f(j \mid A_q) = \sum_{i=1}^{|A_p|} \Delta_f(a_p^i \mid A_q) \leq \sum_{i=1}^{|A_p|} \Delta_f(a_p^i \mid A_q^{(i)}) \leq \sum_{i=1}^{|A_p|} \Delta_f(a_p^i \mid A_p^{(i)}) = f(A_p)$$

The first inequality follows from the submodularity of f, since $A_q^{(i)} \subseteq A_q$. For the second inequality, notice that the algorithm always chooses to allocate an item to the player p with maximal marginal contribution, as long as $f(A_p) < \alpha \cdot \mathrm{OPT}$. □

We will also need an upper bound on $f(A_p)$. The following lemma will give us an upper bound sufficient for the $\frac{1}{3}$-approximation, but we will need a more refined upper bound for the $\frac{2}{5}$-approximation.

Lemma 3. *For all players $p \in P$, $f(A_p) \leq 2 \cdot \alpha \cdot OPT$.*

Proof. Clearly, p can only be allocated one item j that surpasses $\alpha \cdot \mathrm{OPT}$. This item can have size at most $\alpha \cdot \mathrm{OPT}$, by observation 6. Let A' be the partial allocation before giving j to p. It holds that

$$f(A_p) = f(A'_p) + \Delta_f(j \mid A'_p) \leq \alpha \cdot \mathrm{OPT} + f(j) \leq \alpha \cdot \mathrm{OPT} + \alpha \cdot \mathrm{OPT} \leq 2 \cdot \alpha \cdot \mathrm{OPT}$$

□

We can now prove the approximation ratio of $\frac{1}{3}$ by the following lemma.

Lemma 4. $f_{min}(A^*) < f(A_q) + \frac{1}{m} \sum_{p \in P} f(A_p)$.

Proof.

$$\begin{aligned}
f_{\min}(A^*) &= \min_{p \in P} f(A_p^*) && (1)\\
&\leq \min_{p \in P} f(A_p^* \cup A_q) && (2)\\
&\leq \min_{p \in P} f(A_q) + \sum_{j \in A_p^*} \Delta_f(j \mid A_q) && (3)\\
&= f(A_q) + \min_{p \in P} \sum_{j \in A_p^*} \Delta_f(j \mid A_q) && (4)\\
&\leq f(A_q) + \frac{1}{m} \sum_{p \in P} \sum_{j \in A_p^*} \Delta_f(j \mid A_q) && (5)\\
&= f(A_q) + \frac{1}{m} \sum_{p \in P} \sum_{j \in A_p} \Delta_f(j \mid A_q) && (6)\\
&= f(A_q) + \frac{1}{m} \sum_{p \in P - \{q\}} \sum_{j \in A_p} \Delta_f(j \mid A_q) && (7)\\
&\leq f(A_q) + \frac{1}{m} \sum_{p \in P - \{q\}} f(A_p) && (8)\\
&< f(A_q) + \frac{1}{m} \sum_{p \in P} f(A_p) && (9)
\end{aligned}$$

(2) uses monotonicity of f, (3) comes from observation 4. (5) uses the pigeonhole principle, (6) captures the fact that A and A^* allocate the same set of elements. (7) is true because $\Delta_f(j \mid A_q) = 0$ for all $j \in A_q$, (8) uses lemma 2 and (9) holds since $f(A_q) > 0$. $\square$

This indeed implies the approximation ratio for $\alpha = \frac{1}{3}$: We can substitute $\frac{1}{m}\sum_{p \in P} f(A_p) \leq 2 \cdot \alpha \cdot \mathrm{OPT}$ by lemma 3. Taking $2 \cdot \alpha \cdot \mathrm{OPT} = \frac{2}{3} \cdot \mathrm{OPT}$ on the other side, we obtain $\frac{1}{3}\mathrm{OPT} < f(A_q)$.

2.3 Ratio of 2/5

We will now prove that if $\alpha = \frac{2}{5}$, then $f_{\min}(A) \geq \frac{2}{5}\mathrm{OPT}$. To this end, suppose for contradiction that there exists a counterexample to this claim. We will assume that this instance (P, J, f) is minimal with respect to $|P|$ and that f is scaled in such a way that $f_{\min}(A^*) = \mathrm{OPT} = 5$. We also fix a min player q with $f(A_q) = 2 - \varepsilon$ for some $\varepsilon > 0$.

Let us define $\mathrm{avg} := \frac{1}{m}\sum_{p \in P} f(A_p)$. From lemma 4 we obtain $5 = f_{\min}(A^*) < 2 - \varepsilon + \mathrm{avg}$, or, rewritten, $\mathrm{avg} > 3 + \varepsilon$. Our goal is to show that $\mathrm{avg} \leq 3$, arriving at a contradiction.

Let $\beta := \max_{p \in P} \max_{i=2}^{|A_p|} \Delta_f(a_p^i \mid \{a_p^1, ..., a_p^{i-1}\})$ be the largest marginal contribution of an item during the greedy algorithm which is not the first item of a player. If $\beta \leq 1$, then no player can obtain more than value 3, so we can assume $\beta > 1$. Also, $\beta \leq 2$ holds by observation 6.

Lemma 5. *For each set of $k \geq 1$ players $p_1, ..., p_k \in P - \{q\}$, there is no set of k players $r_1, ..., r_k \in P$ such that $\bigcup_{i=1}^k A_{p_i} \subseteq \bigcup_{i=1}^k A^*_{r_i}$.*

Proof. Removing k players from the instance as well as all items from $\bigcup_{i=1}^k A_{p_i}$ leads to a smaller counter-example: The greedy algorithm still behaves identically on all other players, including q, so there is a valid greedy allocation A' on the instance with $f_{\min}(A') = 2 - \varepsilon$. On the other side, since all removed items only belonged to k players in the optimal solution, removing these k players gives a solution A^{**} with $f_{\min}(A^{**}) \geq 5$. $\square$

The above lemma allows us to prove the following:

Lemma 6. *There is a permutation $\phi : P \to P$ such that $A_p \cap A^*_{\phi(p)} \neq \emptyset$ for all $p \in P$.*

Proof. Consider the bipartite graph $G = (P \cup P', E)$ with the set of players P on the left side and a copy P' of the set of players on the right side, and edges (p, p') if $A_p \cap A^*_{p'} \neq \emptyset$. Consider an arbitrary non-empty subset $Q \subseteq P - \{q\}$. By lemma 5, $N(Q) \geq |Q| + 1$. For the set $\{q\}$, we have $N(\{q\}) \geq 1$ since $A_q \neq \emptyset$ and the item in q must be allocated somewhere in the optimal solution. For the set $\emptyset$, clearly $|N(\emptyset)| \geq 0 = |\emptyset|$. For any other set $\{q\} \subset Q \subseteq P$, we know that $N(Q) \geq N(Q - \{q\}) \geq |Q - \{q\}| + 1 = |Q|$. Alltogether, we obtain that for any subset $Q \subseteq P$, $N(Q) \geq |Q|$. We can hence apply Hall's Marriage Theorem to conclude that G contains a perfect matching. This matching clearly defines exactly the desired permutation ϕ. $\square$

To simplify notation, we will consider an optimal solution such that ϕ is the identity, by just permuting the allocations of the players in the optimal solution. We now consider the *reallocation digraph* $G = (P, E)$: Each pair of players $(p, r) \in P^2$ is associated to a set $A_{p \to r} = A_p \cap A^*_r$ which corresponds to the items that were reallocated from p to r, in order to go from A to A^*. An arc (p, r) exists if $p \neq r$ and $A_{p \to r} \neq \emptyset$.

We subdivide the players of our instance into three categories $P = B \cup M \cup S$. The set B of *big* players consists of all players b such that $f(A_b) > 3$. The set M of *medium* players consists

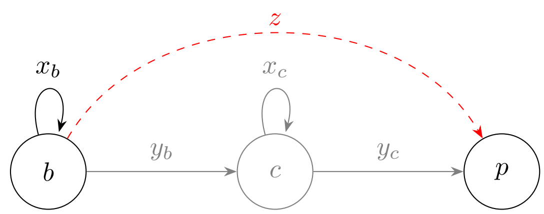


Figure 2: Visualization of lemma 7.

of all players p such that $4 - \beta < f(A_p) \leq 3$. Finally, the set S of *small* players consists of all players p with $f(A_p) \leq 4 - \beta$.

Notice that $q \in S$. This partition is mainly motivated by the following observation:

Observation 7. *For any $p \in B \cup M$, it holds that $|A_p| = 2$.*

Proof. Any player $p \in B \cup M$ cannot be allocated just one item j since $f(j) < 2 \leq 4 - \beta \leq f(A_p)$. But p can also not be allocated more than two items, because in order for greedy to allocate a third item, we need $f(\{a_p^1, a_p^2\}) < 2$, and the first item a_p^1 must have value $f(a_p^1) \geq \beta$. But since marginal contributions are decreasing, this implies that the last item a_p^k can contribute at most $\Delta_f(a_p^2 \mid \{a_p^1\}) \leq 2 - \beta$, and so we have $f(\{a_p^1, a_p^2, ..., a_p^k\}) < 2 + 2 - \beta = 4 - \beta \leq f(A_p)$. □

For each $p \in B \cup M$, denote by x_p the unique item in $A_{p \to p}$ and denote by y_p the other item of A_p. Lemma 6 guarantees the existance of the first item and observation 7 guarantees that there is exactly one other item. Lemma 5 guarantees that the other item is not in $A_{p \to p}$.

Lemma 7. *For any two big or medium players $b \in B \cup M$ and $c \in B \cup M$, $(b, c) \notin E$.*

Proof. Suppose for contradiction that $A_{b \to c} \neq \emptyset$, where $y_b \in A_{b \to c}$. Let $p \in P$ be the player such that $y_c \in A_p^*$. We remove player c as well as items $\{x_c, y_c, y_b\}$ and add an item z which contributes an additive factor of $t := \max(\Delta_f(y_c \mid \{x_c\}), \Delta_f(y_c \mid A_p^* - \{y_c\}))$ to every set. By observation 1 and lemma 1, this function is submodular and monotone. The greedy algorithm on the new instance may allocate all items that give marginal contribution larger than t in the original run as before, because the new instance does not give rise to any new marginal contributions of value $> t$. As soon as all remaining items give marginal contribution $\leq t$, it may choose to allocate item z to player b (since z contributes the same to each player).

Notice that the only player with a potentially different value than in the old run, namely player b, has already received at least value $2 = \alpha \cdot \mathrm{OPT}$: $f(\{x_b, z\}) \geq \beta + \Delta_f(y_c \mid \{x_c\}) \geq \beta + (2 - \beta) = 2$. Therefore, b is ignored by the algorithm. It follows that for the greedy algorithm, the remaining instance is identical to before, so greedy might allocate in the same way. Thus, we again obtain $f_{\min}(A'_q) = 2 - \varepsilon$, for the new allocation A'.

On the other hand, we can modify our old optimal allocation, by just replacing item y_c by item z. Call the new optimal allocation A^{**}. We have

$$f(A_p^{**}) = f(A_p^{**} - \{z\}) + t = f(A_p^* - \{y_c\}) + t \geq f(A_p^* - \{y_c\}) + \Delta_f(y_c \mid A_p^* - \{y_c\}) = f(A_p^*) \geq 5$$

We obtained a counterexample with less players, a contradiction. □

Thus, in fact, $G[B \cup M]$ is an independent set. We now argue that B cannot be too large. The recurrent argument in the following lemmata will be that the players cannot 'spill' too much value which is valuable for player q, because in that case, the item would have been given to player q. For a player p, let us define $V_p := \bigcup_{(r,p)\in E} A_{r\to p}$ the set of items reallocated *to* p and $W_p := \bigcup_{(p,r)\in E} A_{p\to r}$ the set of items reallocated *from* p.

Lemma 8. *For any player p, $\sum_{y\in W_p} \Delta_f(y \mid A_q) < 2$.*

Proof. Let a_p^k be the last item of A_p. By lemma 6, there exists an item $a_p^i \in A_{p\to p}$. Thus,

$$\begin{aligned}
\Delta_f(W_p \mid A_q) &\leq \Delta_f(A_p - \{a_p^i\} \mid A_q) \\
&\leq \sum_{j\neq i} \Delta_f(a_p^j \mid A_q) \\
&\leq \Delta_f(a_p^k \mid \{a_p^1, ..., a_p^{k-1}\}) + \sum_{j\neq i, j\neq k} \Delta_f(a_p^j \mid \{a_p^1, ..., a_p^{j-1}\}) \\
&\leq \Delta_f(a_p^k \mid \{a_p^1, ..., a_p^{i-1}\}) + \sum_{j\neq i, j\neq k} \Delta_f(a_p^j \mid \{a_p^1, ..., a_p^{j-1}\}) \\
&\leq \Delta_f(a_p^i \mid \{a_p^1, ..., a_p^{i-1}\}) + \sum_{j\neq i, j\neq k} \Delta_f(a_p^j \mid \{a_p^1, ..., a_p^{j-1}\}) \\
&= f(A_p - \{a_p^k\}) < 2
\end{aligned}$$

The second inequality is by observation 4. The third inequality is because the items were allocated to p and not q. The next inequality is by definition of submodular functions, then we use that a_p^i was chosen before a_p^k, and finally, the last inequality holds because otherwise a_p^k would not have been allocated to p, since $\alpha \cdot \mathrm{OPT} = 2$. $\square$

Lemma 9. *For any $p \in P$, $\sum_{j\in V_p} \Delta_f(j \mid A_q) \geq 3 + \sum_{j\in W_p} \Delta_f(j \mid A_q) - f(A_p)$.*

Proof.

$$\begin{aligned}
\sum_{j\in V_p} \Delta_f(j \mid A_q) &= \sum_{j\in A_p^*} \Delta_f(j \mid A_q) + \sum_{j\in W_p} \Delta_f(j \mid A_q) - \sum_{j\in A_p} \Delta_f(j \mid A_q) && (10) \\
&\geq \sum_{j\in A_p^*} \Delta_f(j \mid A_q) + \sum_{j\in W_p} \Delta_f(j \mid A_q) - f(A_p) && (11) \\
&\geq \Delta_f(A_p^* \mid A_q) + \sum_{j\in W_p} \Delta_f(j \mid A_q) - f(A_p) && (12) \\
&\geq f(A_p^*) - f(A_q) + \sum_{j\in W_p} \Delta_f(j \mid A_q) - f(A_p) && (13) \\
&\geq 3 + \sum_{j\in W_p} \Delta_f(j \mid A_q) - f(A_p) && (14)
\end{aligned}$$

where (10) comes from the facts $V_p = A_p^* - (A_p - W_p)$, $A_p - W_p \subseteq A_p^*$, and $W_p \subseteq A_p$. (11) comes from lemma 2, (12) is observation 4, (13) follows from $f(A_p^*) - f(A_q) \leq f(A_p^* \cup A_q) - f(A_q) = \Delta_f(A_p^* \mid A_q)$ and (14) is because $f(A_p^*) \geq 5$ and $f(A_q) < 2$. $\square$

Lemma 10. $|B| \leq |S|$.

Proof. Consider the reallocation graph from before, and associate with each player p the quantity $g(p) := \sum_{j \in W_p} \Delta_f(j \mid A_q) - \sum_{j \in V_p} \Delta_f(j \mid A_q)$. Notice that $\sum_{p \in P} g(p) = 0$, since $\bigcup_{p \in P} W_p = \bigcup_{p \in P} V_p$.

By lemma 9, for each player $p \in S \cup M$, we have $g(p) \leq 0$. Now consider a big player $b \in B$, and let p be the player that receives y_b. Since p must be a small player by lemma 7, we have $f(A_p) \leq 4 - \beta$, and by lemma 9, we get $g(p) \leq 1 - \beta$. If we can show $g(b) \leq \beta - 1$, then this implies $g(p) + g(b) \leq 0$. Recall that $W_b = \{y_b\}$. We distinguish two cases.

- If $\Delta_f(y_b \mid A_q) > \beta$, then y_b was the first item that the greedy algorithm gave to b, and thus the second item x_b has value $\leq \beta$. We obtain $f((A_b - W_b) \cup A_q) \leq f(x_b) + f(A_q) < \beta + 2$, and since $f(A_b^*) \geq 5$, we must have

$$\sum_{j \in V_b} \Delta_f(j \mid A_q) \geq \Delta_f(V_b \mid A_q) \geq \Delta_f(V_b \mid (A_b - W_b) \cup A_q) = f(A_b^*) - f((A_b - W_b) \cup A_q) > 3 - \beta.$$

  Therefore, $g(b) = \Delta_f(y_b \mid A_q) - \sum_{j \in V_p} \Delta_f(j \mid A_q) < 2 - (3 - \beta) = \beta - 1$.

- If $\Delta_f(y_b \mid A_q) \leq \beta$, then $f((A_b - W_b) \cup A_q) \leq f(x_b) + f(A_q) \leq 2 + 2$, and since $f(A_b^*) \geq 5$, we must have

$$\sum_{j \in V_b} \Delta_f(j \mid A_q) \geq \Delta_f(V_b \mid A_q) \geq 1.$$

  Therefore, $g(b) = \Delta_f(y_b \mid A_q) - \sum_{j \in V_b} \Delta_f(j \mid A_q) \leq \beta - 1$.

In both cases, we obtain $g(b) + g(p) \leq 0$.

Now consider a player $s \in S$ that receives an item from (at least) two big players b, c. We have $f((A_b - W_b) \cup A_q) \leq f(A_b - W_b) + f(A_q) \leq 4$, but since $f(A_b^*) \geq 5$, we must have $f(V_b) \geq 1$. The same holds for player c. By lemma 8, we also know $g(s) < 2 - \Delta_f(y_b \mid A_q) - \Delta_f(y_c \mid A_q)$. Therefore $g(b) + g(c) + g(s) \leq \Delta_f(y_b \mid A_q) - 1 + \Delta_f(y_c \mid A_q) - 1 + g(s) < 0$.

Now, let S_0 be the set of small players that do not receive any item from big players, let S_1 be the set of small players that receive one item from one big player and let $S_{\geq 2}$ be the set of small players that receive items from multiple big players. Clearly, we can decompose the set of players as

$$P = M \cup S_0 \cup \bigcup_{p \in S_1} \{p, b_1(p)\} \cup \bigcup_{p \in S_{\geq 2}} \{p, b_1(p), b_2(p), ..., b_k(p)\}$$

where $b_i(p)$ denotes the i-th big player that gives an item to p, in an arbitrary order. We have

$$0 = \sum_{p \in P} g(p) = \sum_{p \in M} g(p) + \sum_{s \in S_0} g(s) + \sum_{s \in S_1} g(s) + g(b_1(s)) + \sum_{s \in S_{\geq 2}} g(s) + g(b_1(s)) + ... + g(b_k(s))$$

We showed that all terms are ≤ 0, and that the terms of the last sum are < 0. Hence we obtain $S_{\geq 2} = \emptyset$. However, each big player has an outgoing edge, and by lemma 7, the edge goes to a small player. As $S_{\geq 2} = \emptyset$, no two big players point to the same small player. By pigeonhole principle, $|B| \leq |S|$. $\square$

We can conclude as follows:

$$\begin{aligned}
\text{avg} &= \frac{1}{m}\sum_{p\in P} f(A_p) \\
&\leq \frac{1}{m}(|B|\cdot(2+\beta)+|S|\cdot(4-\beta)+|M|\cdot 3) \\
&\leq \frac{1}{m}(\frac{|B|+|S|}{2}\cdot(2+\beta)+\frac{|B|+|S|}{2}\cdot(4-\beta)+|M|\cdot 3) \\
&= \frac{1}{m}(\frac{6(|B|+|S|)}{2}+|M|\cdot 3) \\
&= 3
\end{aligned}$$

where the last inequality is because $2+\beta \geq 4-\beta$, and $|B| \leq |S|$ by lemma 10. This contradicts avg > 3, as stated in the beginning. We obtain:

Theorem 1. *Truncated Max-Sum Greedy with $\alpha = \frac{2}{5}$ gives a $\frac{2}{5}$-approximation.*

We discuss the (close-to-) tightness of this analysis in Appendix A.2. In particular, we show that no value of α can lead to an approximation factor better than $\frac{43}{105} < 0.41$.

Local search ideas. The main paradigm for constructing our greedy algorithm was to transform a *max-sum* greedy rule into a *max-min* algorithm by a certain truncation of the player value. A similar idea might be applicable for local search procedures. Contrary to greedy, certain bad instances where one player has many copies of an item, but all other players do not have any copy, cease to be possible in the local search case. Therefore, this is a promising direction for improving the ratio of $\frac{2}{5}$.

2.4 Matroid-constrained max-min allocation

We now look at a special case of submodular max-min allocation, related to matroids. Definitions and related concepts as well as the proofs of this section can be found in Appendix A.3.

In this setting, there is only one additive valuation function, but we are given a matroid $\mathcal{M} = (J, \mathcal{I})$, and we must have $A_p \in \mathcal{I}$ for all $p \in P$, i.e. each player can only get an independent set of $\mathcal{M}$. The satisfaction of a player p then equals the weighted rank function of the matroid applied to A_p, which is submodular. This shows that we are indeed dealing with a special case of submodular functions.

The main technical ingredient is an application of the famous *basis exchange property*:

Proposition 2 (Weighted basis exchange property)**.** *Consider a matroid $\mathcal{M} = (J, \mathcal{I})$ and a weight function $w : J \to \mathbb{Q}$. Let B be a non-maximum-weight basis of $\mathcal{M}$. There exist elements $b_1 \notin B$ and $b_2 \in B$ such that $B' := (B - \{b_2\}) \cup \{b_1\}$ is a basis and $w(B') > w(B)$.*

This property allows us to design a simple balancing local search procedure:

Theorem 2. *There is a $\frac{1}{2}$-approximation for max-min allocation subject to a matroid constraint.*

This improves the $\frac{2}{5}$-approximation for general submodular functions, as shown before.

Assignment LP (A)

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j = 1, ..., n$$
$$\sum_{j=1}^{n} f_i'(j) x_{ij} \geq T \qquad \forall i = 1, ..., m$$
$$x_{ij} \geq 0 \qquad \forall i, j$$

Configuration LP (P)

$$\sum_{C \in \mathcal{C}(T,p)} x_{p,C} \geq 1 \qquad \forall p \in P$$
$$\sum_{C \ni j} \sum_{p \in P} x_{p,C} \leq 1 \qquad \forall j \in J$$
$$x_{p,C} \geq 0 \qquad \forall p \in P, C \in \mathcal{C}(T,p)$$

Figure 3: The assignment LP and the configuration LP.

Dual LP (D0)

$$\max \sum_{p \in P} z_p - \sum_{j \in J} y_j \qquad \text{s.t.}$$
$$\sum_{j \in C} y_j \geq z_p \qquad \forall p, C$$
$$y_j, z_p \geq 0 \qquad \forall j, p$$

Dual LP (D1)

$$\sum_{p \in P} z_p > \sum_{j \in J} y_j$$
$$\sum_{j \in C} y_j \geq z_p \qquad \forall p, C$$
$$y_j, z_p \geq 0 \qquad \forall j, p$$

Dual LP (D2)

$$\sum_{j \in J} y_j < m$$
$$\sum_{j \in C} y_j \geq 1 \qquad \forall C$$
$$y_j \geq 0 \qquad \forall j$$

Figure 4: The duals to the configuration LP.

3 Integrality gaps for the configuration LP

In this section, we investigate whether LP rounding approaches, which have proven succesful for similar variants of max-min allocation [5], can be employed in our settings. To this end, we study the integrality gap of the *configuration LP* in the case of identical additive valuation functions. Before defining this LP, let us consider a simpler LP formulation, which we call the *assignment LP*. We want to find the largest T such that the program depicted in figure 3 on the left, where $f_i'(j) = \min(f_i(j), T)$, has a feasible solution. Let us call this LP (A). It is a folklore result that (A) has an integrality gap of 2 when all players have equal valuations.

In the restricted assignment case, (A) has a bad integrality gap [5]. This was the initial motivation to study a stronger LP, namely the aforementioned configuration LP. Let $\mathcal{C}(T,p)$ contain the configurations C such that $f_p(C) \geq T$. We have a variable for each pair of player and configuration. The LP is shown in figure 3 on the right. The largest T such that this LP has a feasible solution is an upper bound to the optimal solution. We will refer to this system as (P).

It is well-known that the intergrality gap of (P) in the general case is $\Omega(\sqrt{m})$ [5], while in the restricted assignment case it was recently shown to be smaller than 3.534 [13]. When valuation functions are submodular, the gap in the restricted assignment case is $\mathcal{O}(\log \log n)$ [4].

(P) has exponentially many variables in general. However, consider its dual, which we call (D0), assuming that the primal minimizes the 0-vector. This LP has polynomially many variables and possibly exponentially many constraints, and is depicted in figure 4 on the left.

The separation problem for this LP is the classical Knapsack problem. Since Knapsack admits a (F)PTAS [20], we can solve the LP up to any accuracy using the Ellipsoid method.

3.1 Dual Fitting for the configuration LP

Notice that in (D0), a trivial solution always exists by setting all variables to 0. But if we had a solution with objective value larger than 0, then we could obtain an arbitrary high objective value by scaling all variables. By LP duality, we get that if (D0) is unbounded, then (P) with the same threshold T is infeasible. Additionally, we can directly express the requirement that the objective function must be greater than 0 as a constraint. We call the resulting LP (D1), and it is shown in the middle of figure 4.

This is the LP which we will consider when players have different valuations. In the case of equal valuations, we can without loss of generality set all player variables to 1, and obtain (D2), as seen on the right of figure 4. This LP searches a fractional hitting set of size smaller than m that hits all configurations larger than our threshold value. In (D1) and (D2), we call the first set of constraints *value constraints* and the second set of constraints *configuration constraints*.

If (D2) or (D1) has a feasible solution, then (D0) is unbounded, and by weak duality, (P) is infeasible. We thus have the following dual fitting framework:

Proposition 3. *Consider a class Γ of max-min allocation instances, and denote by $OPT(\mathcal{I})$ the optimal integral value of instance $\mathcal{I}$. If for all $\mathcal{I} \in \Gamma$, (D2) or (D1) is feasible for $\mathcal{I}$ and threshold value $\alpha \cdot OPT(\mathcal{I})$, then the integrality gap of the configuration LP on Γ is at most α.*

Let us strengthen the notion of optimality a bit. For this section, an allocation A is *optimal* if $f_{\min}(A)$ is maximized and among all allocations with maximal $f_{\min}(A)$, the number of players achieving exactly $f_{\min}(A)$ is minimized. The following easy lemma will be very helpful:

Lemma 11. *Consider an instance with identical additive valuations. Assume there is a set of k disjoint configurations $C_1, ..., C_k$ of value $\geq T$ strictly contained in the allocation of k players, $A_{\ell_1} \cup ... \cup A_{\ell_k}$, with $T > f_{\min}(A)$. Then, A is not optimal.*

Proof. Let $j^* \in \bigcup_{i=1}^{k} A_{\ell_i} - \bigcup_{i'=1}^{k} C_{i'}$. Reallocate j^* to the min player, and for all i', allocate $C_{i'}$ to player $\ell_{i'}$. The resulting allocation has higher value or contains one min player less. □

When invoking this lemma, we will often just specify j^* and $C_1, ..., C_k$.

Throughout this section, we will use the notation $j_p := \arg\max_{j \in A_p} f(j)$ to refer to the item with the largest weight in A_p. We will refer as j'_p to the second largest item, j''_p to the third largest, and so on. We now have all necessary tools to prove upper bounds on the integrality gap of the configuration LP. We start by a rather simple and general result.

Lemma 12. *For an optimal integral allocation A, let T be the largest value obtained by a player p with $|A_p| > 1$. The configuration LP on $\mathcal{I}$ is infeasible for any value larger than T.*

Proof. We can assume $n > m$, otherwise we can set the smallest item j to $y_j = 1 - \varepsilon$ and all other items j' to $y_{j'} = 1$, and (D2) is feasible for any value larger than OPT.

For an item $j \in A_p$, define $y_j := \frac{f(j)}{f(A_p)}$. Notice that $\sum_{j \in J} y_j = m$, so this is not yet a feasible solution to (D2). Consider an arbitrary configuration C with value larger than T. If C contains an item j which is allocated in a singleton configuration $A_p = \{j\}$, then $y_j \geq \frac{f(j)}{f(A_p)} = 1$ and so the constraint corresponding to C is satisfied.

Otherwise, we have $\sum_{j \in C} y_j \geq \sum_{j \in C} \frac{f(j)}{T} > \sum_{j \in C} \frac{f(j)}{f(C)} = 1$. Thus, no configuration constraints of a variable of a non-singleton item j are tight. Such j must exist since $n > m$.

We can hence lower y_j by a sufficiently small value $\zeta > 0$, ensuring that still all configuration constraints are satisfied and furthermore, $\sum_{j \in J} y_j = m - \zeta < m$. We conclude that (D2) is feasible, and thus for any $\varepsilon > 0$, the configuration LP is infeasible for value $T + \varepsilon$. $\square$

This already gives an upper bound of 2 to the integrality gap with identical additive valuations, since any allocation A where $T > 2 \cdot f_{\min}(A)$ can be improved by a simple reallocation to the min player. We now extend this result to *parallel machines*, which we define below.

Parallel machines. We are given a single additive valuation function f, but additionally, each player p has a *speed* s_p. The valuation of player p for a set A_p is then $\frac{f(A_p)}{s_p}$. Notice that if $s_p = 1$ for all $p \in P$, we recover the identical valuations setting.

The biggest challenge in generalizing the above result to the parallel machines setting is that items which satisfy a player alone (i.e. $j \in A_p$ with $|A_p| = 1$) are not necessarily large anymore. Therefore, a plain assignment of item sizes to the y-variables of (D1) and of player speeds to the z-variables of (D1) fails. However, by carefully shrinking down the variable values of items given to players with larger speeds, we still prove the desired result in Appendix B.1.

Proposition 4. *For an instance $\mathcal{I}$ in the parallel machines setting, the configuration LP on $\mathcal{I}$ is infeasible for any value larger than $2 \cdot OPT$.*

Another result about parallel machines, which we prove in Appendix B.1, is the following:

Proposition 5. *Consider an instance with two players p, q in the parallel machines setting. The integrality gap of the configuration LP is* 1.

Submodular valuations. Finding a constant upper bound to the integrality gap of the configuration LP with identical submodular valuations remains open. However, in Appendix B.2, we prove the following lower bound and also discuss potential approaches to prove upper bounds.

Proposition 6 (Lower bound on the integrality gap)**.** *The integrality gap of the configuration LP with identical submodular valuations is at best $4/3$, even when $n = 6$ and $m = 3$.*

In the following, we present the technically most involved proof using our dual fitting method.

3.2 Equal additive valuations

The main theorem of this section states:

Theorem 3. *The integrality gap of the configuration LP with equal valuations is at most $\frac{3}{2}$.*

The full proof is given in Appendix B.3. We give a short synopsis in the following.

We assume by normalization $f_{\min}(A) = 2$ for the optimal integral allocation A. Our goal is to give a satisfying assignment to (D2). The idea is to assign y-weights to the items such that $\sum_{j \in A_p} y_j = 1$ for all $p \in P$. We then argue how we can lower a variable by a sufficiently small amount, to indeed satisfy $\sum_{j \in J} y_j < m$. For players who receive at least 4 items in A, we want to give a special value to the heaviest item, and distribute the remaining value among all other items proportionally. Therefore, these other items should not be too heavy, which is the essence of the following lemma. Recall that j_p is the largest item of player p.

Lemma 13. *If $|A_p| \geq 4$, then $f(A_p - \{j_p\}) < 1$.*

Now we describe the assignment of y-variables. Let t be the player that maximizes j_t over all t with $|A_t| = 3$ under the condition $j_t < 1$ (if such a player exists). We define a threshold $T := \frac{8}{3} - j_t$. Consider a player p. If $|A_p| = 1$, set $y_j = 1$ for the unique $j \in A_p$. Now suppose $|A_p| = 2$. If $j'_p \leq 1$ or $j_p \geq T$, set $y_{j_p} = \frac{2}{3}$ and $y_{j'_p} = \frac{1}{3}$. Otherwise, set $y_j = \frac{1}{2}$ for both $j \in A_p$.

Now consider $|A_p| = 3$. If $j_p \geq T$, set $y_{j_p} = 2/3$, and set the other two items proportionally, i.e. $y_j = \frac{j}{3 \cdot f(A_p - \{j_p\})}$. If $T > j_p \geq 1$, set $y_{j_p} = 1/2$, and $y_j = \frac{j}{2 \cdot f(A_p - \{j_p\})}$. If $j_p < 1$, set, for all three $j \in A_p$, $y_j = 1/3$. Finally, when $|A_p| \geq 4$, if $j_p \geq T$, set $y_{j_p} = 2/3$, else set $y_{j_p} = 1/2$. Set the rest of the items proportionally, as above.

With this assignment of y-variables, the main remaining work to prove the theorem is to show that the configuration constraints are satisfied, which comes down to many case distinctions:

Lemma 14. *Let C be a configuration with $f(C) > 3$. Then, $\sum_{j \in C} y_j \geq 1$.*

3.3 A Lower bound to the integrality gap and Petersen instances

For the min-max version of item allocation, a classical scheduling problem, it was shown in [17] that the integrality gap is at least $1 + \frac{1}{1023}$. Their example is based on a construction using the Petersen graph, which we denote by Π.

Definition 1 (Petersen instance with respect to a vertex labelling)**.** *Let $x \in \mathbb{Z}^{10}$ be a vertex labelling of Π. The Petersen instance with respect to x has three players and one item j_e for each $e \in E(\Pi)$. The weight of an item $j_{\{u,v\}}$ is $x_u + x_v$.*

Π contains exactly 6 perfect matchings, and any vertex is covered by exactly two of them. Therefore, taking each of the six perfect matching configurations to an extent of $\frac{1}{2}$ satisfies (P) with value $\sum_{v \in V(\Pi)} x_v$. However, since Π does not contain three disjoint perfect matchings (in particular, Π is not *1-factorizable*), it is impossible to obtain a balanced integral allocation, for certain choices of x. The authors prove this for the labelling $x = (2^0, ..., 2^9)$.

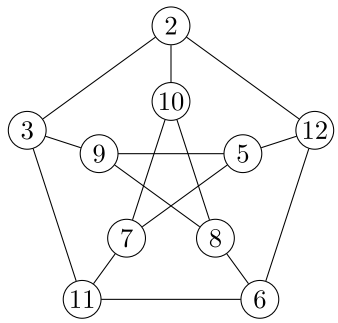
The same example shows a $1 + \frac{1}{1022}$ lower bound for the max-min version. However, a smaller x could lead to a better bound for the integrality gap. Indeed, we experimentally found a better labelling of the vertices, depicted in figure 5. The vertex labels sum to 73, and one can computationally check that no integral allocation gives value 73 to each of the three players. So, the best integral allocation achieves at most 72, but the fractional allocation described above gives value 73. We obtain:

Theorem 4. *The integrality gap of the configuration LP with identical additive valuations is at least $1 + \frac{1}{72}$. This holds already with only 5 different item sizes.*

This result also holds for the min-max allocation setting and therefore improves the lower bound on the integrality gap of [17] from $1 + \frac{1}{1023}$ to $1 + \frac{1}{73}$.

For a fixed $\varepsilon > 0$, let us define a *worst-case instance* as an instance which achives the worst possible integrality gap up to an additive ε. We use ε for purely technical reasons, since a worst-case instance might not be defined for $\varepsilon = 0$. In the following, we want to show that under certain assumptions, the Petersen instance is a worst-case instance. This might be an indication that the true integrality gap might be closer to the bound of theorem 4 than to the bound of theorem 3, since the Petersen instances seem to capture the "essence" of the integrality gap, at least under some assumptions. All proofs can be found in Appendix B.4.

Figure 5: The Petersen graph Π with an improved vertex labelling.



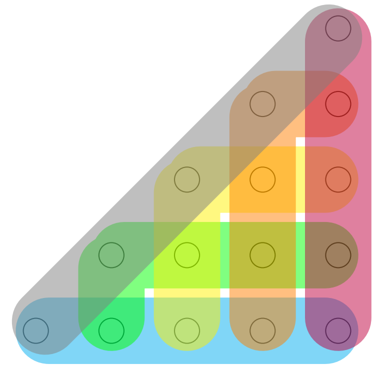


Figure 6: The hypergraph H.

Proposition 7. *Consider an instance with $m \leq 3$ and restrict the support of the configuration LP to configurations of equal value, and to size at most 6. Then, there exists a Petersen instance which is a worst-case instance.*

Notice that if $m < 3$, then proposition 5 implies that the integrality gap is 1.

For the proof, consider an instance $\mathcal{I}$ where the optimal solution to (P) has value T, and for the optimal integral solution, $\mathrm{OPT}(\mathcal{I}) < T$. Let $G = (V, E)$ be a graph where V are the configurations in the support of the optimal solution to (P), and two configurations are connected if they share an item.

Lemma 15. $G \cong K_6$.

Lemma 16. *There exists a worst-case instance with exactly* 15 *items.*

Let us consider a hypergraph H where each item is a vertex and each configuration is a hyperedge. We have $|V(H)| = 15$ and $|E(H)| = 6$. Since we know by lemma 16 that each configuration intersects every other configuration in exactly one item, H is uniquely defined, and depicted in figure 6. When considering worst-case instances, we can restrict ourselves to configurations corresponding to $E(H)$. Our only degree of freedom is to choose the weights of the items, in such a way that all configurations sum to the same value. Let Z be the vector space of all such weight assignments to $V(H)$. By the above arguments, the instances from Z must contain a worst-case instance.

Let X be the 10-dimensional vector space of all weight assignments of $V(\Pi)$, the vertices of the Petersen graph. Let Y be the vector space of all weight assignments of the 15 edges of Π which correspond to a Petersen instance.

Lemma 17. *There is a bijection between X and Y.*

This immediately implies that Y is also a 10-dimensional vector space.

Lemma 18. $Y \subseteq Z$.

Lemma 19. $dim(Z) = 10$.

This essentially follows from the fact that we can describe Z as the solution space of a system of linear equations with 15 variables and 5 linearly independent constraints.

But now we know $\dim(Y) = \dim(Z)$, and $Y \subseteq Z$. This implies $Y = Z$. Since the instances from Z contain a worst-case instance, so do the instances from Y. This concludes the proof.

4 Constrained agents

We now divide the set of players into two categories: The set of *threshold players* P_t and the set of *sum players* P_{sum}. Our goal is to find an allocation that gives to each threshold player at least their capacity and maximizes the average value that a sum player receives.

However, as we will show in observation 8, there is no hope for finding an approximation algorithm of any factor for the problem: The constraint of satisfying the threshold players is already too strong, independently of the maximization of the sum players. Therefore, we will resort to so-called *bi-criteria approximation algorithms.* The idea is to find an allocation such that each min player receives at least an α-fraction of her capacity, and the average value allocated to the sum players is at least β of the optimal allocation respecting the constraints of the threshold players. More formally, we define the following promise problem:

CONSTRAINED ITEM ALLOCATION
Input: a set of items J, a set of players $P = P_t \cup P_{\text{sum}}$, capacities C_i for each $i \in P_t$, a valuation function f and an integer k
Promise: There is an A^* with $f(A_i^*) \geq C_i$ for all $i \in P_t$, and $\sum_{i \in P_{\text{sum}}} f(A_i^*) \geq k$.
Output: An (α, β)-approximation to A^*, i.e., an allocation A such that $f(A_i) \geq \alpha \cdot C_i$ for all $i \in P_t$ and $\sum_{i \in P_{\text{sum}}} f(A_i) \geq \beta \cdot k$

Observation 8. *Unless* $\boldsymbol{P} = \boldsymbol{NP}$*, there is no* $(1, \beta)$*-approximation algorithm for* CONSTRAINED ITEM ALLOCATION*, for any* $\beta > 0$*, even for* 2 *players, additive valuations and equal capacities.*

Proof. In fact, just finding a *feasible* solution of the described special case without any sum player is exactly the PARTITION problem, which is **NP**-hard. □

4.1 Parallel and additive

We will first consider the case where the valuation functions are additive, but the capacities of the threshold players are potentially different. Observe that in this case, we can merge all sum players to one threshold player with capacity k. Our problem now reduces to finding an assignment of items that achieves an as good as possible ratio to the given capacities.

The first greedy algorithm that comes to mind is the one that orders the items by non-increasing size, and then assigns each item in order to the player who is missing the most (absolute) value. We refer to this algorithm as *absolute greedy.* We first show why this algorithm has no useful performance guarantee, which motivates another simple greedy algorithm, which we call *relative greedy*: Items are assigned to the player such that after the allocation, the relative coverage of the capacity of the player is minimized. A pseudocode can be seen in algorithm 2.

Proposition 8. *The absolute greedy has no performance guarantee, even in the case of only two different capacities, one of them appearing only once.*

Algorithm 2: The relative greedy

Data: items J with weight function $w : J \to \mathbb{Q}$, players P, capacities $c : P \to \mathbb{Q}$
Result: an allocation A
for $j_1, ..., j_n \in J$ *ordered by non-increasing* $f(j_i)$ **do**
  $q \leftarrow \arg\min_{p \in P} \frac{A_p + f(j_i)}{C_p}$;
  $A_q \leftarrow A_q \cup \{j_i\}$;
end
return $(A_1, ..., A_m)$

On the other hand, relative greedy overcomes this bad example:

Theorem 5. *Relative greedy gives a* $\frac{1}{2}$*-approximation.*

4.2 Equal constraints

In the case where all threshold players have the same capacity C, we can be more ambitious and look for a $(\alpha, 1)$-approximation. Letting W denote the total weight of all items, our problem reduces to finding a set of items of weight at most $W - k$ such that all threshold players receive at least C. A natural candidate for such an approximation is the well-known *longest processing time* (LPT) greedy algorithm. In our case, we will only use the largest items summing to at most $W - k$, and we will possibly ignore very large (of size $> C$) items, since they should be allocated to the sum players. The guessing operation only adds a linear factor to the running

Algorithm 3: The capped LPT greedy

Data: items J, additive function $f : J \to \mathbb{Q}$, threshold players P_t, sum player s, min capacity C, sum threshold k
Result: an allocation A
$B \leftarrow \{j \in J \mid f(j) > C\}$;
Guess $\ell \in \{0, ..., |B|\}$;
Allocate the ℓ smallest items (call this set L) of B to ℓ threshold players;
$J \leftarrow J - B$;
$A_s \leftarrow B - L$;
while $J \neq \emptyset \wedge \exists j \in J : f(j) \leq f(A_s) + f(J) - k$ **do**
  $j \leftarrow \arg\max_{j \in J : f(j) \leq f(A_s) + f(J) - k} f(j)$;
  $q \leftarrow \arg\min_{p \in P_t} f(A_p)$;
  $A_q \leftarrow A_q \cup \{j\}$;
end
return $(A_1, ..., A_m)$

time. We call this algorithm *capped LPT greedy*, and prove the following in Appendix C.2:

Theorem 6. *Capped LPT Greedy gives a* $(\frac{1}{2}, 1)$*-approximation.*

Conclusion. Due to space constraints, the reader is referred to the end of page 2.

References

[1] C. Annamalai, C. Kalaitzis, and O. Svensson, *Combinatorial algorithm for restricted max-min fair allocation*, ACM Transactions on Algorithms (TALG), 13 (2017), pp. 1–28.

[2] A. Asadpour, U. Feige, and A. Saberi, *Santa claus meets hypergraph matchings*, ACM Transactions on Algorithms (TALG), 8 (2012), pp. 1–9.

[3] Y. Azar and L. Epstein, *Approximation schemes for covering and scheduling in related machines*, Approximation Algorithms for Combinatiorial Optimization: International Workshop APPROX'98 Aalborg, Denmark, July 18–19, 1998 Proceedings 1, (1998), pp. 39–47.

[4] É. Bamas, S. Morell, and L. Rohwedder, *The submodular santa claus problem*, Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), (2025), pp. 616–640.

[5] N. Bansal and M. Sviridenko, *The santa claus problem*, Proceedings of the thirty-eighth annual ACM symposium on Theory of computing, (2006), pp. 31–40.

[6] I. Bezáková and V. Dani, *Allocating indivisible goods*, ACM SIGecom Exchanges, 5 (2005), pp. 11–18.

[7] D. Chakrabarty, J. Chuzhoy, and S. Khanna, *On allocating goods to maximize fairness*, 2009 50th Annual IEEE Symposium on Foundations of Computer Science, (2009), pp. 107–116.

[8] S. Chaudhary, R. Ramjee, M. Sivathanu, N. Kwatra, and S. Viswanatha, *Balancing efficiency and fairness in heterogeneous gpu clusters for deep learning*, Proceedings of the Fifteenth European Conference on Computer Systems, (2020), pp. 1–16.

[9] C. Chekuri, J. Vondrák, and R. Zenklusen, *Dependent randomized rounding via exchange properties of combinatorial structures*, 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, (2010), pp. 575–584.

[10] S. Davies, T. Rothvoss, and Y. Zhang, *A tale of santa claus, hypergraphs and matroids*, Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, (2020), pp. 2748–2757.

[11] M. X. Goemans, N. J. Harvey, S. Iwata, and V. Mirrokni, *Approximating submodular functions everywhere*, Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms, (2009), pp. 535–544.

[12] D. Golovin, *Max-min fair allocation of indivisible goods*, School of Computer Science, Carnegie Mellon University, (2005).

[13] P. Haxell and T. Szabó, *Improved integrality gap in max–min allocation, or, topology at the north pole*, Combinatorica, 45 (2025), pp. 1–38.

[14] S. Khot, R. J. Lipton, E. Markakis, and A. Mehta, *Inapproximability results for combinatorial auctions with submodular utility functions*, Algorithmica, 52 (2008), pp. 3–18.

[15] A. Krause, R. Rajagopal, A. Gupta, and C. Guestrin, *Simultaneous placement and scheduling of sensors*, 2009 International Conference on Information Processing in Sensor Networks, (2009), pp. 181–192.

[16] D. Kurokawa, A. D. Procaccia, and N. Shah, *Leximin allocations in the real world*, ACM Transactions on Economics and Computation (TEAC), 6 (2018), pp. 1–24.

[17] A. Kurpisz, M. Mastrolilli, C. Mathieu, T. Mömke, V. Verdugo, and A. Wiese, *Semidefinite and linear programming integrality gaps for scheduling identical machines*, Mathematical Programming, 172 (2018), pp. 231–248.

[18] V. Mirrokni, M. Schapira, and J. Vondrák, *Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions*, Proceedings of the 9th ACM conference on Electronic commerce, (2008), pp. 70–77.

[19] L. Poláček and O. Svensson, *Quasi-polynomial local search for restricted max-min fair allocation*, ACM Transactions on Algorithms (TALG), 12 (2015), pp. 1–13.

[20] S. Sahni, *Approximate algorithms for the 0/1 knapsack problem*, Journal of the ACM (JACM), 22 (1975), pp. 115–124.

[21] J. Vondrák, *Optimal approximation for the submodular welfare problem in the value oracle model*, Proceedings of the fortieth annual ACM symposium on Theory of computing, (2008), pp. 67–74.

[22] G. J. Woeginger, *A polynomial-time approximation scheme for maximizing the minimum machine completion time*, Operations Research Letters, 20 (1997), pp. 149–154.

Appendix A Submodular max-min allocation

Appendix A.1 Bad example for the Natural Greedy Algorithm

Each of our items cover a certain amount of ground area of the plane. Consider n base items $B_1, ..., B_n$ and a fixed $\varepsilon < 1$. For any set S of items, we set its value to be $f(S) = |\bigcup_{s \in S} s|$, that is the amount of area it covers. For every $q \in \mathbb{Q}$ and every set B, fix an arbitrary partition of B into sets of size q, as well as an ordering over the elements of that partition. Denote by $B^{p,q,\delta}$ a set containing the p-th part of the partition of B into sets of size q, as well as a unique part of size δ (this part does not appear in any other set).

Now, we will describe m item groups $G_1, ..., G_m$, which will correspond to a possible greedy allocation. For this, we first define some quantities for each i, namely ε_i (the *gain part*), s_i, r_i (the *repeat part*) and t_i. Intuitively, ε_i corresponds to the marginal contribution of an item of G_i added to player i, s_i is the size of B_i, r_i describes how much of an item in G_i was already covered by player i, and t_i is the number of items in G_i.

Let $\varepsilon_1 := \varepsilon/2m$, $s_1 := 1$, $r_1 := \frac{1}{2}$, and $t_1 := 2m$. Now, for each $i \in \{2, ..., m\}$, we set $s_i := s_{i-1} + (t_{i-1} - 1) \cdot \varepsilon_{i-1}$, $r_i := \frac{\varepsilon_{i-1}}{2}$, $t_i := \frac{m}{r_i}$ and $\varepsilon_i := \frac{\varepsilon_{i-1}}{t_i}$. By the inductive definition of s_i, we have:

Observation 9. *The sequence $s_1, ..., s_m$ is increasing.*

For each i, we put the following items into G_i: A *base item* $B_i^{1,1,0}$, as well as m copies of $B_i^{j,r_i,\varepsilon_i}$ for each $j \in \{1, ..., r_i\}$, which we call *fractional items.* Notice that these fractional items cover m times a fraction of size 1 of the corresponding base item.

The greedy algorithm clearly will start by allocating one base item to each player, and without loss of generality, for the rest of the proof, we will assume that $B_i^{1,1,0}$ has been allocated to player i. We start with some simple lemmata.

Lemma 20. *For each fractional item $j \in G_i - \{B_i^{1,1,0}\}$, $f(j) < \varepsilon_{i-1}$.*

Proof. The size of an item is equal to its repeat part (r_i) and its gain part (ε_i). Thus,

$$f(j) = r_i + \varepsilon_i = r_i + \frac{r_i \cdot \varepsilon_{i-1}}{m} < 2r_i = \varepsilon_{i-1}$$

since $\frac{\varepsilon_{i-1}}{m} \leq \frac{\varepsilon_1}{m} < 1$. □

Lemma 21. *$f(G_j) = 1 + \varepsilon$ for all $j \in P$.*

Proof. By induction over j. For player 1, $f(G_1) = s_1 + t_1 \cdot \varepsilon_1 = 1 + 2m \cdot \frac{\varepsilon}{2m} = 1 + \varepsilon$. If it holds for $j - 1$, then we have

$$\begin{aligned} f(G_j) &= s_j + t_j \cdot \varepsilon_j = s_j + \varepsilon_{j-1} = s_{j-1} + (t_{j-1} - 1) \cdot \varepsilon_{i-1} + \varepsilon_{j-1} \\ &= s_{j-1} + t_{j-1} \cdot \varepsilon_{j-1} = f(G_{j-1}) = 1 + \varepsilon \end{aligned}$$

□

Lemma 22. *Fix a player i. Consider a partial allocation $A = (A_1, ..., A_m)$ with the following properties:*

- *For all $j < i$, $A_j = G_j$*

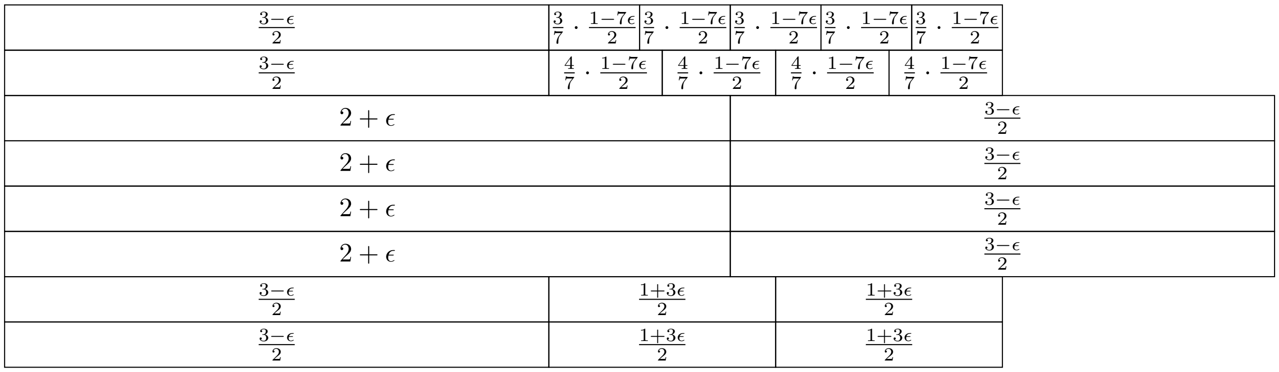


Figure 7: A possible greedy allocation of our instance.

- *$A_i \subset G_i$*
- *For all $j > i$, $A_j = \{B_j^{1,1,0}\}$*

Then, $f(A_i) = f_{min}(A)$.

Proof. By lemma 21, $f(A_j) = 1 + \varepsilon$ for all $j < i$. Since each fractional item of G_i gives gain ε_i, we have $f(A_i) \leq s_i + (t_i - 1) \cdot \varepsilon_i < s_i + \varepsilon_{i-1} = s_{i-1} + (t_{i-1} - 1) \cdot \varepsilon_{i-1} + \varepsilon_{i-1} = s_{i-1} + t_{i-1} \cdot \varepsilon_{i-1} = f(A_{i-1}) = 1 + \varepsilon$. On the other hand, for any $j > i$, by observation 9, $f(A_j) = s_j \geq s_{i+1} = s_i + (t_i - 1) \cdot \varepsilon_i \geq f(A_i)$. Therefore, A_i is minimal among $A_1, ..., A_m$. $\square$

With the above lemmata, it is clear that the following is a valid run of Greedy: First assign $B_i^{1,1,0}$ to each player i. Now by lemma 22, player 1 has minimum value. By submodularity of f and lemma 20, any fractional item from G_j for some $j > 1$ must contribute less to A_1 than any fractional item from G_1. This together with lemma 22 implies that Greedy can choose to allocate all fractional items of G_1 to player 1. One more time by lemma 22, the next lowest player is 2, and we can allocate all items of G_2 to player 2 etc., until all items of G_i are allocated to player i for all $i \in \{1, ..., m\}$. We obtain the allocation $A = (G_1, ..., G_m)$. By lemma 21, each player obtains value $1 + \varepsilon$.

However, another allocation would be to allocate to each player its base item and for all fractional items, each distinct item once. Notice that each fractional item appears n times, so this is a valid allocation. More formally, we consider $A^* = (A_1^*, ..., A_m^*)$ with

$$A_i = \{B_i^{1,1,0}\} \cup \{B_j^{\ell, r_j, \varepsilon_j} \mid j \in [m], \ell \in [r_j]\}$$

Since in each group G_i, the fractional items cover a B_i-fraction of size 1, in total at least size m is covered for each player. Hence, we obtain a gap of $\frac{1+\varepsilon}{m}$, for any $\varepsilon > 0$.

Appendix A.2 Tight example for the truncated max-sum greedy

In the following, we describe an instance which shows that the truncated max-sum greedy cannot lead to an approximation factor better than $\frac{43}{105} \approx 0.4095$. Suppose the threshold $\alpha \cdot \mathrm{OPT}$ is $2+\varepsilon$

for a tiny $\varepsilon > 0$. Player q is allocated an item of size $2+\varepsilon$ and there are m copies of sufficiently small fractions of that item. The idea is that these fractional items will also be allocated to q by the greedy algorithm, because they are handled last, but they are sufficient to give an additive value of $2+\varepsilon$ to each player in an optimal allocation. Consider a partial instance that gives rise to the following greedy allocation (also depicted in figure 7):

- two players get $\{\frac{3-\varepsilon}{2}, 2 \times \frac{1+3\varepsilon}{2}\}$.
- four players get $\{2+\varepsilon, \frac{3-\varepsilon}{2}\}$.
- one player gets $\{\frac{3-\varepsilon}{2}, 4 \times \frac{4}{7} \cdot \frac{1-7\varepsilon}{2}\}$.
- one player gets $\{\frac{3-\varepsilon}{2}, 5 \times \frac{3}{7} \cdot \frac{1-7\varepsilon}{2}\}$.

In the optimal allocation, we can assume that each player has an additional item of size $2+\varepsilon$, coming from player q (who is not part of the partial instance). We can have the following configurations:

- four players get $\{2+\varepsilon, 2 \times \frac{3-\varepsilon}{2}\}$
- four players get $\{2 \times 2+\varepsilon, \frac{1+3\varepsilon}{2}, \frac{4}{7} \cdot \frac{1-7\varepsilon}{2}, \frac{3}{7} \cdot \frac{1-7\varepsilon}{2}\}$.

The configurations in the optimal allocation sum to 5. We used 8 additional items of size $2+\varepsilon$ from the min player, but we used one item of size $\frac{3}{7} \cdot \frac{1-7\varepsilon}{2}$ less. Therefore, we can repeat this partial instance sufficiently often. The greedy allocate will still behave in the same way, and eventually, we will have enough free items of size $\frac{3}{7} \cdot \frac{1-7\varepsilon}{2}$ such that player q achieves value 5 in the optimal allocation.

For the greedy allocation to be valid, we need that all items except the last sum to at most $2+\varepsilon$. This is easy to see for the first two configurations. For the last two configurations, we must have

$$\frac{3-\varepsilon}{2} + 4 \cdot \frac{3}{7} \cdot \frac{1-7\varepsilon}{2} = \frac{3-\varepsilon}{2} + 3 \cdot \frac{4}{7} \cdot \frac{1-7\varepsilon}{2} \leq 2+\varepsilon$$

which is satisfied for $\varepsilon = \frac{1}{21}$. Therefore, we found an instance where greedy only achieves $\frac{2+\varepsilon}{5} = \frac{43}{105} \approx 0.4095$.

We gave an example when $\alpha \geq \frac{43}{105}$. When $\alpha < \frac{43}{105}$, then it is easy to construct an instance where only a $\frac{43}{105}$-fraction of the optimal solution is achieved, since the greedy algorithm does not guarantee anything when all players have value larger than $\alpha \cdot \mathrm{OPT}$.

Appendix A.3 Matroid-constrained max-min allocation

We give the definition of a matroid and two useful equivalent characterizations.

Definition 2 (Matroid). *A* matroid *is a pair* $\mathcal{M} = (J, \mathcal{I} \subseteq 2^J)$ *with*

(I1) $\emptyset \in \mathcal{I}$

(I2) if $A \in \mathcal{I}$*, then* $A' \in \mathcal{I}$ *for all* $A' \subseteq A$

(I3) if $A, B \in \mathcal{I}$ *and* $|A| > |B|$*, then* $\exists x \in A - B$ *with* $B \cup \{x\} \in \mathcal{I}$*.*

We denote by $M[J']$ the matroid M restricted to the subset $J' \subseteq J$. The *rank function* of a matroid maps a subset S of elements to the cardinality of the largest independent set contained in S. A *base* of a matroid is a maximal independent set.

Equivalent characterizations. Let $\mathcal{B}$ be the set of maximal independent sets of a matroid. The *basis exchange property* states that for any two distinct $A, B \in \mathcal{B}$ and $a \in A - B$, there exists a $b \in B - A$ such that $(A - \{a\}) \cup \{b\} \in \mathcal{B}$. Together with the requirement $\mathcal{B} \neq \emptyset$, this is an alternative characterization of matroids.

Another characterization is the following: Let $\mathcal{I}$ be a family of sets on the ground set J, closed under taking subsets, where each ground element is associated to a weight. Consider the algorithm which greedily chooses the maximum-weight element still available in J, as long as the constructed set is still in the set system. It holds that $(J, \mathcal{I})$ is a matroid if and only if this greedy algorithm produces a maximum-weight independent set under any weight assignment.

Proposition 2 (Weighted basis exchange property)**.** *Consider a matroid $\mathcal{M} = (J, \mathcal{I})$ and a weight function $w : J \to \mathbb{Q}$. Let B be a non-maximum-weight basis of $\mathcal{M}$. There exist elements $b_1 \notin B$ and $b_2 \in B$ such that $B' := (B - \{b_2\}) \cup \{b_1\}$ is a basis and $w(B') > w(B)$.*

Proof. Consider the classical greedy algorithm, which chooses the maximum-weight element which keeps the current set independent, and specify a tiebreak rule in case where two elements have the same weight as follows: Items in B are prioritized. Now let B^* be the maximum-weight basis produced by this greedy algorithm. We have $B \neq B^*$ (otherwise B is maximum-weight). By the basis exchange property, there exists $x \in B - B^*$ and $y \in B^* - B$ such that $B - \{x\} \cup \{y\}$ is a basis in $\mathcal{M}$.

Consider the moment where the greedy algorithm chooses y. Since $x \notin B^*$, item x is still available. As y is chosen, we have $w(y) \geq w(x)$. Furthermore, since $x \in B$ but $y \notin B$, if $w(x) = w(y)$, then the greedy algorithm would have chosen x. Thus, $w(y) > w(x)$. We can conclude $w((B_2 - \{x\}) \cup \{y\}) > w(B_2)$. □

Theorem 2. *There is a $\frac{1}{2}$-approximation for max-min allocation subject to a matroid constraint.*

Proof. Let A be an optimal solution with $f_{\min}(A) = 1$, and assume that for all $j \in J$, we have $f(j) \leq 1$. We first show that $f_{\max}(A) \leq 2$, and afterwards we derive a simple balancing procedure which achieves a $\frac{1}{2}$-approximation.

Suppose that there is a player p with $f(A_p) > 2 \cdot f(A_q) = 1$. Since A is optimal, it must hold that $A_q \cup \{j\} \notin \mathcal{I}$ for all $j \in A_p$. Otherwise, we could reallocate j from p to q.

Now, let us restrict $\mathcal{M}$ to $J' := A_q \cup A_p$. Call this restriction $\mathcal{M}' = (J', \mathcal{I}')$. If there was some $j \in J' - A_q$ such that $A_q \cup \{j\} \in \mathcal{I}'$, then we can reallocate j from p to q and achieve a better solution. Therefore, such an element does not exist, and by definition, A_q is a basis of $\mathcal{M}'$. We know that A_p is independent in $\mathcal{M}$; therefore, we also have $A_p \in \mathcal{I}'$. So, there exists a basis in $\mathcal{M}'$ with weight at least $f(A_p) > f(A_q)$. This implies that A_q is a non-maximum-weight basis in $\mathcal{M}'$. By proposition 2, there exists $x \in A_q$ and $y \in J' - A_q = A_p$ such that $f(y) > f(x)$ and $S := (A_q - \{x\}) \cup \{y\} \in \mathcal{I}'$, and thus also $S \in \mathcal{I}$. We get that $f(S) > f(A_q)$. Additionally, we have $f(A_p) - \{y\} > 2 \cdot f(A_q) - f(A_q) = f(A_q)$. We can hence reallocate y from p to q, and obtain a better allocation, a contradiction.

Now consider the following algorithm: Guess the optimum, and assume by scaling that it has value 2. Start from an arbitrary allocation A. If $f_{\min}(A) \geq 1$, we are done, since $f_{\min}(A^*) \leq f_{\text{sum}}(A^*)/m < 2 \leq 2 \cdot f_{\min}(A)$. Otherwise, there must be a player p with $f(A_p) > 2$ (since the average value of a player must be at least 2). If A_q is not a basis of $\mathcal{M}[A_q \cup A_p]$, there exists $j \in A_p$ such that $A_q \cup \{j\} \in \mathcal{I}$. In this case, reallocate j from p to q. Otherwise, iteratively perform swaps of two items x, y between p and q as described above, until $f(A_p) < 2$ or $f(A_q) > 1$.

Repeat this procedure until $f_{\min}(A) \geq 1$. This must eventually happen, since the average value of the players is 2. In every iteration, at least one player permanently obtains value between 1 and 2, and thus there can be at most $\mathcal{O}(m)$ iterations.

□

Appendix B Configuration LP

Appendix B.1 The configuration LP on parallel machines

Proposition 4. *For an instance $\mathcal{I}$ in the parallel machines setting, the configuration LP on $\mathcal{I}$ is infeasible for any value larger than $2 \cdot OPT$.*

Proof. Among all optimal allocations, fix an allocation A that minimizes $\beta_A := \max_{p \in P} \frac{f(A_p)}{s_p}$. By normalization, we can assume $f(A_q)/s_q = 1$ for the min player q. Let P^* be the set of players such that $|A_p| = 1$ and $f(A_p)/s_p > 2$. Items allocated to a player in P^* are called *special*, while all other items are called normal. We describe an algorithm that assigns y-values to the items. Order the players by nondecreasing speed. Set an initial scaling factor $\eta = 1$. Every time the scaling factor is changed, we will say that a new *phase* begins, and we will denote the η-value of the t-th phase by $\eta^{(t)}$. Additionally, let us say that an item j is *big* if $f(j) \geq f(j')$ for all items j' from previous phases. The other items are called *small*. In the first phase, all items are big.

For the next player p, set $z_p = s_p \cdot \eta$. If $p \in P^*$, set $y_j = z_p$ for the unique $j \in A_p$ and update $\eta \leftarrow 2 \cdot \frac{z_p}{f(j)}$. If $p \notin P^*$, partition the items as follows. Let A'_p be all big items of A_p (except for one item, in case all items are big). The rest set $A''_p := A_p - A'_p$ consists of either one big item of size $< s_p$ or small items of size $< s_t$, where t is any player of the previous phase. This is because otherwise, we could swap j_t and A''_p and obtain an optimal allocation with smaller β_A-value. We have $f(A'_p) < s_p$, otherwise we can reallocate A''_p to A_q (and symmetrically, $f(A''_p) < s_p$). We set $y_j = \frac{f(j)}{2 \cdot f(A^*_p)} \cdot z_p$ for all $j \in A^*_p$ for $* \in \{',''\}$.

For the value constraint, notice that

$$\sum_{j \in J} y_j = \sum_{p \in P - P^*} \sum_{j \in A_p} y_j + \sum_{p \in P^*} \sum_{j \in A_p} y_j = \sum_{p \in P - P^*} \sum_{j \in A_p} \frac{f(j)}{f(A_p)} \cdot z_p + \sum_{p \in P^*} z_p = \sum_{p \in P} z_p.$$

Therefore, if no configuration constraint is tight, we can lower an arbitrary variable to make the value constraint feasible. We now turn to the configuration constraints:

Consider a player p from phase t and a configuration C with $f(C)/s_p > 2$. We want to show $\sum_{j \in C} y_j > z_p$. If C contains a big item $j \in A_r$ from a later phase t', then

$$y_j = f(j)/f(A'_r) \cdot s_r \cdot \eta^{(t')}/2 = f(j)/f(A'_r) \cdot s_r/f(j_r) \cdot z_p \geq \frac{s_r \cdot z_p}{f(A'_r)} > z_p,$$

and so j alone satisfies this configuration. The same holds for big items from A''_r. We can therefore focus on configurations without big items from later phases.

Now consider the small items of later phases, C_{small}. Let u be the last player of the current phase. We have

$$\sum_{j \in C_{\text{small}}} y_j \geq \sum_{r \in P} \sum_{j \in A_r \cap C_{\text{small}}} \frac{f(j)}{f(A''_r)} \cdot z_t \cdot \frac{s_r}{f(j_u)} > \frac{f(C_{\text{small}}) \cdot z_u \cdot s_r}{s_p \cdot f(j_u)} = \frac{f(C_{\text{small}})}{s_p} \cdot z_u > \frac{f(C_{\text{small}})}{f(C)} \cdot z_p$$

Now consider special items of previous phases, C^*. We get

$$\sum_{j \in C^*} y_j = \sum_{j \in C^*} z_p \geq \sum_{j \in C^*} \frac{f(j) \cdot \eta^{(t)}}{2} > \frac{f(C^*) \cdot z_p}{2 \cdot s_p} > \frac{f(C^*)}{f(C)} \cdot z_p$$

For C', the normal items of the same phase t or a previous phase, we have

$$\sum_{j \in C'} y_j \geq \sum_{j \in C'} f(j) \cdot \frac{s_p}{f(A_p)} \cdot \eta^{(t)} \geq \sum_{j \in C'} \frac{f(j)}{2} \cdot \eta^{(t)} = \frac{f(C')}{2} \cdot \eta^{(t)} = \frac{f(C') \cdot z_p}{2 \cdot s_p} > \frac{f(C')}{f(C)} \cdot z_p$$

In total, $\sum_{j \in C} y_j > \frac{f(C^*)+f(C')+f(C_{\text{small}})}{f(C)} \cdot z_p = z_p$ and thus, (D1) is feasible. □

Proposition 5. *Consider an instance with two players p, q in the parallel machines setting. The integrality gap of the configuration LP is* 1*.*

Proof. For each $j \in J$, set $y_j = f(j)$. Let q be the player such that $f(A_q)/s_q < f(A_p)/s_p$. We want to show that the (D1) is feasible for $\frac{f(A_q)+\varepsilon}{s_q}$, for any $\varepsilon > 0$. Set $z_q = f(A_q) + \varepsilon$ and $z_p = f(A_p)$.

There cannot be any configuration C with $\frac{f(A_q)+\varepsilon}{s_q} \leq f(C)/s_p < f(A_p)/s_p$. Otherwise, giving C to player p and $J - C$ to player q leads to a better allocation.

Consider any configuration C which is valid for q, that is $f(C) \geq f(A_q) + \varepsilon$. We have $\sum_{j \in C} y_j = f(C) \geq f(A_q) + \varepsilon$, and the corresponding constraint is satisfied.

Now, consider any configuration C which is valid for p, that is $f(C)/s_p \geq \frac{f(A_q)+\varepsilon}{s_q}$. As argued before, we must have $f(C) \geq f(A_p)$. Thus, $\sum_{j \in C} y_j = f(C) \geq f(A_p)$, and the constraint is satisfied.

The value constraint is satisfied since $z_p + z_q - \sum_{j \in J} y_j = f(A_p) + f(A_q) + \varepsilon - f(J) = f(J) + \varepsilon - f(J) > 0$. □

Appendix B.2 Submodular valuation functions

Proposition 6 (Lower bound on the integrality gap)**.** *The integrality gap of the configuration LP with identical submodular valuations is at best $4/3$, even when $n = 6$ and $m = 3$.*

Proof. Consider items $j_1, j_2, j_3, k_1, k_2, k_3$ and a submodular function defined as $f(j_a) = f(k_a) = 2$ for all a, $f(\{j_a, j_b\}) = f(\{k_a, k_b\}) = 4$ for all $a \neq b$, $f(\{j_a, k_b\}) = 3$ and $f(S) = 4$ for all S such that $|S| \geq 3$. Our instance has 3 players. For all $j \in J$, we have $\Delta_f(j \mid S) = 2$ when $|S| = 0$, $\Delta_f(j \mid S) \in [1, 2]$ when $|S| = 1$, $\Delta_f(j \mid S) \in [0, 1]$ when $|S| = 2$ and $\Delta_f(j \mid S) = 0$ when $|S| > 2$, which implies submodularity of f (monotonicity is clear).

The configuration LP is feasible for this instance with threshold value 4: We set $x_{p,\{j_a,j_b\}} = x_{p,\{k_a,k_b\}} = 1/6$ for all p and $a \neq b$. Each player gets one unit of configuration (because there are six configurations in the support) and each item appears only in two configurations of the support, and thus in at most one unit of configuration.

Now, assume that there is an integral solution of value at least 4. Since there are six items and three players, and a singleton item only gets value 2, each player must be allocated exactly 2 items. But in any allocation of configurations of two items to three players, one player must obtain items of the form $\{j_a, k_b\}$, which only gives value 3. □

Remark 1 (Hypergraph matching techniques)**.** *In the restricted assignment case, the technique which has proven most successful consists in reducing the problem to a certain version of finding relaxed matchings in hypergraphs. The vertices represent items and the hyperedges represent configurations. Even though this technique can in principle be applied to our setting, this does not seem to be particularly promising: The restriction to certain hyperedges (i.e. configurations) is necessary in the restricted assignment case, but in the identical-valuations case, such a graph structure is far too general.*

We conjecture the following:

Conjecture 1. *The integrality gap of the configuration LP with equal submodular valuations is $\mathcal{O}(1)$.*

Let A be an optimal allocation. For a player p, let $\Pi_{A,p}$ denote the set of all permutations of items in A_p.

In particular, we believe that a promising assignment of y-variables is, for each $j \in A_p$,

$$y_j := \frac{1}{f(A_p) \cdot |A_p|!} \sum_{\pi \in \Pi_{A,p}} \Delta_f(y \mid \{\pi(i) \mid i < \pi^{-1}(j)\})$$

This corresponds to the expected marginal contribution when adding an item $j \in A_p$ in a random order.

Appendix B.3 Additive valuation functions

Suppose that there exists an optimal allocation A with $f_{\min}(A) = 2$, and an $\varepsilon > 0$ such that the configuration LP is feasible for threshold $3 + \varepsilon$. Let q be a min player, i.e. $f(A_q) = 2$. If no player with at least two items achieves value more than 3, then lemma 12 already allows us to arrive at a contradiction. Therefore, we can assume that there exists a player r with $f(A_r) > 3$ and $|A_r| \geq 2$. In fact, since we must have $f(A_r - \{j\}) \leq 2$ for any $j \in A_r$ (otherwise reallocate j to the min player), it is easy to see that $|A_r| = 2$. Let $A_r = \{x, y\}$ with $x \geq y$ be the two items allocated to player r.

We start with a useful lemma:

Lemma 13. *If $|A_p| \geq 4$, then $f(A_p - \{j_p\}) < 1$.*

Proof. First let us assume $j_p < 0.5$. Let S be a minimal set with $f(S) \geq (2 - x)$. We know $f(S) < \max(j_p, 2(2-x))$. But then, $f(A_p - S) > \max(2 - f(j_p), 2 - 2(2-x)) \geq 1$. This implies $|A_p - S| \geq 3$. We can find a strict subset $S' \subset A_p - S$ containing value at least value

$$\frac{2}{3} \cdot f(A_p - S) \geq \frac{2}{3} \cdot (2 - (2(2-x))) = \frac{4x-4}{3} > \frac{4x-4}{4} = x - 1 = 2 - (3 - x) > 2 - y$$

But then we can invoke lemma 11 with $C_1 = S \cup \{x\}$, $C_2 = S' \cup \{y\}$ and $j^* \in A_p - S - S'$. So we can assume $j_p \geq 0.5$.

Now assume for contradiction $f(A_p - \{j_p\}) \geq 1$.

- Case 1: There exists an item $j \in A_p$ with $j \leq 2 - x$. Let $S := A_p - \{j_p, j\}$. By assumption we have $f(S) \geq 1 - f(j) \geq x - 1 > 2 - y$. Apply lemma 11 with $C_1 = \{j_p, x\}$, $C_2 = S \cup \{y\}$ and $j^* = j$.

- Case 2: All items have size $> 2 - x$. Let j^-, j^{--} be the two smallest items in A_p. Invoke lemma 11 with $C_1 = \{x, j^-\}$, $C_2 = A_p - \{j^-, j^{--}\}$ and $j^* = j^{--}$. We have $f(\{y\} \cup A_p - j^- - j^{--}) > 1 + \frac{f(A_p)}{2} \geq 2$, and also $f(\{x, j^-\}) > f(x) + 2 - f(x) = 2$.

□

Lemma 23. *If there exists a player t with $|A_t| = 3$ and $y_t < 1$, then for all p with $A_p = \{j_p, j'_p\}$ with $j_p \geq \frac{8}{3} - j_t$, we have $j'_p \leq \frac{4}{3}$.*

Proof. Otherwise, we can invoke lemma 11 with $C_1 = \{j'_p, j_t\}$, $C_2 = \{j_p, j'_t\}$, both summing to more than 2, and $j^* = j''_t$. □

Lemma 14. *Let C be a configuration with $f(C) > 3$. Then, $\sum_{j \in C} y_j \geq 1$.*

Proof. We call items set to 1 *singleton*, items set to 2/3 *big*, items set to 1/2 *medium* and items set to 1/3 *small*. All other items are called *normal items*. Using lemma 13 and the above definitions, one can verify that for all normal items j, we have $y_j \geq f(j)/2$.

Consider an arbitrary configuration C with value larger than 3. We want to show $\sum_{j \in C} y_j \geq 1$. If C contains a singleton item, a big and a medium/small item, two medium items, or three small items, we are done. Now we consider all remaining cases:

- C contains only normal items. In this case, $\sum_{j \in C} y_j \geq \frac{1}{2} \sum_{j \in C} f(j) > 3/2 > 1$.

- C contains one big or medium item j and normal items. We have $f(C - \{j\}) > 3 - 2 = 1$, so we get $\sum_{j' \in C} y_{j'} \geq y_j + f(C - \{m\})/2 > \frac{1}{2} + \frac{1}{2} = 1$.

- C contains one small item s and normal items. We have $f(s) \leq 4/3$, by lemma 23, and therefore $f(C - \{s\}) > 3 - 4/3 = 5/3$. Thus, $\sum_{j \in C} y_j \geq y_s + f(C - \{s\})/2 > \frac{1}{3} + \frac{5}{6} > 1$.

- C contains two small items s, s' and normal items. If $f(\{s, s'\}) \leq 7/3$, then we are done, since the rest 2/3 contributes at least 1/3 value, and s, s' contribute $\frac{2}{3}$ value.

  Suppose for contradiction that s, s' are allocated to p, p' with $|A_p| = |A_{p'}| = 2$. Notice that $p \neq p'$ since there can only be one small item per such player. $\{s, s'\}$ have value more than 2, as argued above; $\{j_p, j_t\}$ have value more than two, by definition; and $\{j_{p'}, j'_t\}$, have value more than two: $j_{p'} + j'_t \geq 8/3 - j_t + (2 - j_t)/2 = 11/3 - 3j_t/2 > 2$. We can apply lemma 11 with $C_1 = \{s, s'\}$, $C_2 = \{j_p, j_t\}$, $C_3 = \{j_{p'}, j'_t\}$ and $j^* = j''_t$.

  Thus, C can contain at most one small item from a 2-item-player. This item can have weight at most 4/3 by lemma 23. The other small item can have weight at most 1 by definition. Therefore, $f(C - \{s, s'\}) > 3 - 1 - \frac{4}{3} = \frac{2}{3}$. We get

  $$\sum_{j \in C} y_j = y_s + y_{s'} + f(C - \{s, s'\})/2 > 1/3 + 1/3 + 1/3 = 1$$

  .

- C contains one medium item m and one small item s. Let $S = A_p - \{m\}$, where $m \in A_p$. We must have $f(S) \geq 1$. If $m > 4/3$, then invoke lemma 11 with $C_1 = \{m, j_t\}$, $C_2 = \{y\} \cup S$, $C_3 = \{x, j'_t\}$ and $j^* = j''_t$. This is possible since $m + j_t > 4/3 + 2/3 = 2$, $f(\{y\} \cup S) > 1 + 1 = 2$, and $x + j'_t > 3/2 + 1/2 = 2$. Otherwise, we have $f(C - \{m, s\}) > 3 - \frac{4}{3} - \frac{4}{3} = 1/3$, and we have $\sum_{j \in C} y_j \geq y_m + y_s + f(C - \{m, s\})/2 > 1/2 + 1/3 + 1/6 = 1$.

□

So, all configuration constraints of the dual LP are satisfied. To make the value constraint feasible, we must lower an arbitrary variable by some $\varepsilon > 0$.

- If a normal item exists, we can lower the variable of this item, since normal items are not involved in any tight constraints.
- If no normal item exists, then we consider the min player q. If $A_q = \{j\}$ for a singleton item j, we can clearly lower y_j. Otherwise, q has either a big item of value at most $4/3$, or a small item of value at most $2/3$.
  - Suppose q has a big item b with $f(b) \leq \frac{4}{3}$. The only case where b can be involved in a tight constraint is in a configuration with one small item s. But since $f(s) \leq \frac{4}{3}$ by lemma 23, such a configuration can have value at most $y_b + y_s \leq \frac{4}{3} + \frac{4}{3} \leq 3$, and is thus not considered.
  - Suppose q has a small item s with $f(s) \leq \frac{2}{3}$. The only case where s can be involved in a tight constraint is in a configuration with one big item b (but then $y_s + y_b \leq \frac{2}{3} + 2 \leq 3$) or with two other small items s', s'' (but then $y_s + y_{s'} + y_{s''} \leq \frac{2}{3} + \frac{4}{3} + 1 \leq 3$).

We conclude that (D2) is feasible, and thus by proposition 3, the integrality gap is shown.

Appendix B.4 Petersen instances

Lemma 15. $G \cong K_6$.

Proof. We first claim that G must be a clique. Indeed, if there were $C \neq C'$ with $\{C, C'\} \notin E$, by our assumption on the size of the configurations in (P), we know $f(J - C - C') \geq T$. Thus, there are three disjoint configurations $C, C', J - C - C'$ which all sum to at least T, contradicting the value of $\mathrm{OPT}(\mathcal{I})$.

Since G is a clique, any two configurations can sum to at most one unit of configuration, so the total configuration can sum to at most $|V|/2$. We get $|V| \geq 6$, and by assumption $|V| = 6$. □

Lemma 16. *There exists a worst-case instance with exactly* 15 *items.*

Proof. Suppose that there exists an item that appears in three configurations. These three configurations can sum up to at most 1 by the LP constraints. Since the other three configurations also share items pairwise, they can only sum to at most 1.5. In total, in that case we can have at most 2.5 units of configuration, which is not sufficient for three players. Therefore, each item appears in exactly two configurations, which implies that each item corresponds to one edge of G. This also implies that our instance has at least 15 items.

If our worst-case instance had more than 15 items, there must exist a pair of configurations C, C' such that at least two items j, j' appear only in these two configurations. Consider the modified instance where we remove j, j' and add an item j'' with $f(j'') = f(j) + f(j')$. The optimal integral solution can clearly only become worse. On the other hand, replacing $\{j, j'\}$ by $\{j''\}$ in both C and C' gives a solution to the configuration LP of the same value. □

Lemma 17. *There is a bijection between X and Y.*

Proof. Let $f : X \to Y$ be defined as $f(x)_{\{u,v\}} := x_u + x_v$. The proof of injectivity essentially follows from the fact that every vertex of the Petersen graph is involved in an odd cycle: Suppose for contradiction that there are vectors $x \neq x'$ such that $f(x) = f(x')$. Consider a vertex v_1 such that $x_{v_1} = x'_{v_1} + \varepsilon$ for some $\varepsilon > 0$, and any incident edge $e = \{v_1, v_2\}$. There must be a cycle $(v_1, v_2, v_3, v_4, v_5, v_1)$. Since $f(x)_e = f(x')_e$, we must have $x_{v_2} = x'_{v_2} - \varepsilon$. Iterating the same argument, we must have $x_{v_3} = x'_{v_3} + \varepsilon$, $x_{v_4} = x'_{v_4} - \varepsilon$, and $x_{v_5} = x'_{v_5} + \varepsilon$. But then, for $e' = \{v_5, v_1\}$, we must have $f(x)_{e'} = f(x')_{e'} + 2\varepsilon$, and thus $f(x) \neq f(x')$. □

Lemma 18. *$Y \subseteq Z$.*

Proof. Let Φ be the hypergraph with the edges of Π as vertices and perfect matchings of Π as hyperedges. It is a well-known property of Π that each perfect matching intersects each other perfect matching in exactly one edge. But this implies $\Phi \cong H$.

Now let $y \in Y$, and let $x \in X$ be the corresponding assignment of $V(\Pi)$ such that $y_{\{u,v\}} = x_u + x_v$. Assign to each vertex value z_v of H the corresponding edge weight from y. Since any $e \in E(\Phi)$ corresponds to a perfect matching of Π, we get $\sum_{v \in e} z_v = \sum_{v \in V(\Pi)} x_v$ for each $e \in E(H)$, and therefore all configurations sum to the same value. □

Lemma 19. *$dim(Z) = 10$.*

Proof. Z is the solution set for the following system of linear equations:

$$\sum_{v \in C_i} y_v = \sum_{v \in C_{i+1}} y_v \qquad \forall 1 \leq i \leq 5$$

This system has five independent linear constraints and 15 variables. Suppose for contradiction that they are not linearly independent. Let i be an index such that the i-th constraint is linearly dependent on the other constraints. If $i = 5$, we can set all vertices of C_5 to 1, and all other vertices to 0: This implies that $C_1 = ... = C_4 = 1$, but $C_5 = 5$, and therefore all constraints but i are satisfied, a contradiction.

Now, suppose $i \leq 4$. For all $j \leq i$, and vertices v such that $v \in C_j$, set $y_v = 0$. For all other vertices v, set $y_v = 1$. By definition, $\sum_{v \in C_{j-1}} y_v = 0 = \sum_{v \in C_j} y_v$ for all $j \leq i$. Also, for all $j > i + 1$, we have $\sum_{v \in C_{j-1}} y_v = 6 - i - 1 = \sum_{v \in C_j} y_v$. Therefore, all other constraints are fulfilled. However, $0 \neq 6 - i - 1$, and therefore the i-th constraint is violated, a contradiction.

We can conclude that the vector space of the solution set of the linear equation system has dimension $|V(H)| - 5 = 10$. □

Appendix C Constrained Agents

Appendix C.1 Relative greedy

Before turning to greedy algorithms, we remark that results in the literature already imply a $(1-\varepsilon, 1-\varepsilon)$-approximation algorithm for the problem: If we regard the sum player as just another threshold player, our problem reduces to simple max-min allocation on parallel machines, which admits a PTAS [3]. However, a simpler greedy algorithm is still of interest.

We first analyze the absolute greedy. In the following, the restriction in the following proof of unboundedness is interesting, since it implies that absolute greedy is not applicable even in the case of equal capacities:

Proposition 8. *The absolute greedy has no performance guarantee, even in the case of only two different capacities, one of them appearing only once.*

Consider three players p_1, p_2, p_3 with capacity $C = 2$ and one player p_4 with capacity $C_4 = \varepsilon$. We could for example choose $\varepsilon = \frac{1}{8}$. We have the following items: Three items of size 1, two items of size $1 - \varepsilon$, one item of size $\frac{1}{2}$, one item of size $\frac{1}{2} - \varepsilon$, one item of size 2ε and two items of size ε. In this order, the absolute greedy could distribute the items as follows: p_1, p_2 and p_3 first all get an item of size 1. Then, p_2 and p_3 get an item of size $1 - \varepsilon$. Then, p_1 gets the items of size $\frac{1}{2}$ and $\frac{1}{2} - \varepsilon$. Now, each player misses a value of ε, so p_1 might get the item of size 2ε, p_2 and p_3 might get the items of size ε, and p_4 remains without an element.

On the other side, one can check that the allocation $(A_1 = \{1-\varepsilon, 1-\varepsilon, 2\varepsilon\}, A_2 = \{1, 1\}, A_3 = \{1, \frac{1}{2}, \frac{1}{2} - \varepsilon, \varepsilon\}, A_4 = \{\varepsilon\})$ gives to each player exactly their capacity. Now, we consider the relative greedy.

Theorem 5. *Relative greedy gives a $\frac{1}{2}$-approximation.*

Proof. Consider a greedy allocation A. Let q be the player that minimizes $A_q/C_q =: \alpha$. Assume for contradiction that $\alpha < \frac{1}{2}$. We distinguish two cases.

First, assume there is no player r with $|A_r| = 1$ and $f(A_r) > C_r$. In that case, we can show that $f(A_p) \leq C_p$ for all players p: For a fixed p, consider the first item j allocated to p. If no item is allocated to p, trivially $f(A_p) = f(\emptyset) \leq C_p$. If $f(j) > \alpha \cdot C_p$, then relative greedy will never assign another item to p, and since by assumption $f(j) \leq C_p$, we conclude $f(A_p) \leq C_p$.

Otherwise, consider the last item z that finishes before $\alpha \cdot C_p$. Denote by A' the partial allocation just before z was allocated. Clearly, $f(A'_p) \leq \alpha \cdot C_p$. Also, player p can only get one more item y allocated. By the non-increasing order of our elements, we have $f(y) \leq \alpha \cdot C_p$. We conclude that $f(A_p) = f(A'_p) + f(y) \leq 2\alpha \cdot C_p < C_p$. The total weight of all items is at most

$$W \leq \alpha \cdot C_q + \sum_{p \in P - \{q\}} C_p = \sum_{p \in P} C_p - \alpha \cdot C_q = W - \alpha \cdot C_q < W$$

resulting in a contradiction.

Now, consider the last time in the relative greedy algorithm where a player p who receives a single item j with $f(j) > C_p$. All items that were allocated up to now are larger than C_p, and can be allocated only to players with larger capacity in any optimal allocation. Thus, the sum of all items allocated until now cannot exceed the sum of the capacities of all players with larger capacity, that is, $\sum_{y \in A'} f(y) \leq \sum_{j \in P} C_j$. Also, for all players r with $C_r > C_p$, it must hold that $\frac{f(A'_r)+f(j)}{C_r} \geq \frac{f(j)}{C_p} > \frac{1}{2} > \frac{f(A_q)}{C_q}$, by the rule of the relative greedy algorithm. Therefore no item will be allocated to A'_r anymore, i.e. $A'_r = A_r$. Alltogether, the total weight of all items is

$$\begin{aligned}
W &\leq \sum_{r \in P: C_r > C_p} f(A_r) + \sum_{r \in P: C_r \leq C_p} f(A_r) \\
&\leq \sum_{r \in P: C_r > C_p} C_r + \alpha \cdot C_q + \sum_{r \in P - \{q\}: C_r \leq C_p} C_r \\
&\leq \sum_{r \in P} C_r - \alpha C_q < \sum_{r \in P} C_r = W
\end{aligned}$$

where in the second inequality, for all players r with $C_r \leq C_p$, we can apply the above result that $f(A_r) \leq C_r$. This leads again to a contradiction, finishing the proof. □

Appendix C.1.1 Tightness

Consider a fixed $\varepsilon > 0$ and items of weight $1-\varepsilon, 1-2\varepsilon, ..., \frac{1}{2}+\varepsilon$, as well as $N := \frac{1}{2\varepsilon}$ many items of size $\frac{1}{2}$, and N many players, player i having capacity $C_i = 1.5 - (i+1)\varepsilon$, for all $i \in [N]$.

The greedy algorithm might first allocate one item of weight $1 - k\varepsilon$ to player k for each k. Let A' be the partial allocation at this point. The remaining items are $N-1$ items of size $\frac{1}{2}$. Notice that for any $i < N$, $\frac{f(A'_i)+0.5}{C_i} = \frac{1.5-i\varepsilon}{1.5-(i+1)\varepsilon} > \frac{1.5-N\varepsilon}{1.5-(N+1)\varepsilon} = \frac{f(A'_N)+0.5}{C_N}$. Therefore, the $N-1$ items will be allocated to all players except N. It follows that $f(A_N) = 1 - N\varepsilon = \frac{1}{2}$, and thus player N achieves only a $\frac{1}{2-2\varepsilon}$-fraction of her capcacity. Also notice that all other players $i \in \{1, ..., N-1\}$ achieve $f(A_i) = 1 - i\varepsilon + \frac{1}{2} = 1.5 - i\varepsilon = C_i + \varepsilon$.

On the other hand, consider the allocation A^* which is obtained from A by giving the first item that player i obtained to player $i-1$, for all $i \in \{2, ..., N\}$, and giving the first item of player 1 to player N. For any player $i \in \{1, ..., N-1\}$, we have $f(A_i^*) = f(A_i) + (1-(i+1)\varepsilon) - (1-i\varepsilon) = f(A_i) - \varepsilon = C_i$. Player N gives away her only item and gets the biggest item, thus $f(A_N^*) = 1 - \varepsilon = C_N$. We conclude that each player gets exactly her capacity. For ε going to 0, relative greedy produces a solution which is only half-optimal, and therefore the above theorem is tight.

Appendix C.2 Capped LPT Greedy

Theorem 6. *Capped LPT Greedy gives a $(\frac{1}{2}, 1)$-approximation.*

Let ℓ_{opt} be the number of items in B allocated to threshold players in an arbitrary optimal allocation. We show the following simple lemma:

Lemma 24. *There exists an optimal allocation that allocates the ℓ_{opt} smallest items of B to threshold players (and no other items from B).*

Proof. Each threshold player that receives an item from B is satisfied. Therefore, replacing an item j from B by a smaller y item from B does not impact the fact that the corresponding threshold player achieves value at least C. On the other side, the value of s cannot decrease, since if y was allocated to s, now we can allocate j to s, which has larger value. $\square$

We can now proceed with the proof of the main theorem.

Proof. Let A be an allocation by our greedy algorithm with the guess $\ell = \ell_{\text{opt}}$. It is clear that the algorithm will never allocate an item if the items remaining for s sum to less than k, and by that $f(A_s) \geq k$. So it remains to show that for any player $p \in P_t$, we have $f(A_p) \geq \frac{C}{2}$.

We can disregard the ℓ threshold players that got an item from L, since their allocations are in bijection with the allocation to ℓ players in an optimal solution, by lemma 24. After deleting these players from our instance, we will now assume that no item j with $f(j) > C$ got allocated to a threshold player in our algorithm.

We distinguish two cases. First, assume that in the greedy allocation A, there is a player $q \in P_t$ with $f(A_q) > C$. A_q cannot consist of a single item, since all items j with $f(j) > C$ were allocated to player s. Let z be the last item allocated to q. In order for z to be allocated to q, all other players must have value at least $f(A_q - \{z\})$. But by the order of the algorithm, we have $f(A_q - \{z\}) \geq \frac{f(A_q)}{2} > \frac{C}{2}$.

In the second case, we have $f(A_p) \leq C$ for all players $p \in P_t$. Consider the player q who gets allocated the minimum value $f(A_q) = C - t$ for some $t \leq C$. By the optimality of $\ell_{\text{opt}} = \ell$, we must have $f(J - B) \geq C \cdot |P_t|$ (because an optimal solution must cover each threshold player by at least C). Denote by J_{final} the set J at the end of the algorithm. If $J_{\text{final}} = \emptyset$, then

$$C \cdot |P_t| \leq f(J - B) = f(A_q) + \sum_{p \in P_t} f(A_p) \leq f(A_q) + C \cdot |P_t - \{q\}|$$

which implies $f(A_q) \geq C$, so the solution of the algorithm is optimal. We can hence assume $J_{\text{final}} \neq \emptyset$, and take an arbitrary element $j \in J_{\text{final}}$. As the while condition was false at the end of the algorithm,

$$\begin{aligned}
f(j) &> f(A_s) + f(J_{\text{final}}) - k \\
&= W - \sum_{p \in P_t} f(A_p) - k \\
&= W - (f(A_q) + \sum_{p \in P_t - \{q\}} f(A_p)) - k \\
&\geq W - (C - t) - |P_t| \cdot C - k \\
&\geq |P_t| \cdot C + k + t - |P_t| \cdot C - k \\
&= t
\end{aligned}$$

But since the algorithm allocates items in order of non-increasing size, all items y that have been allocated to q must also satisfy $f(y) > t$. So, $f(A_q) > t$. Together with our assumption $f(A_q) = C - t$, we conclude $f(A_q) > \frac{C}{2}$.

□

Tightness We will present two kinds of worst-case instances. One where greedy manages to choose a subset of items of size exactly $W - k$, and one where greedy manages to allocate the chosen subset of items optimally. In both cases, we will show that greedy still does not have a performance better than $\frac{1}{2}$.

Example 1. *Fix an $\varepsilon > 0$. Consider $m = \frac{1+\varepsilon}{2\varepsilon}$ players, $2m - 1$ items of size $1 + \varepsilon$, and m items of size $1 - \varepsilon$. Let $W - k = 2m$. The greedy algorithm will allocate one item of size $1 + \varepsilon$ to each player, and another item of size $1 + \varepsilon$ to each player except for one. After that, we will have allocated items of weight $(2m - 1)(1 + \varepsilon) = \frac{1+\varepsilon}{\varepsilon} = 2m$, so our greedy algorithm will terminate with one player getting value $1 + \varepsilon$.*

However, giving to each player one item of size $1 + \varepsilon$ and one item of size $1 - \varepsilon$ is also a valid allocation and gives minimum value 2.

Example 2. *Consider two threshold players, one item of size 2, one item of size $1 + \varepsilon$, and 4 items of size 1, with $W - k = 4$. Greedy will select the item of size 2 and the item of size $1 + \varepsilon$, and give one item to each player, achieving $f_{min}(A) = 1 + \varepsilon$. This allocation is optimal with respect to the chosen subset of items.*

On the other hand, one can allocate the four items of size 1 equally to the two players, achieving $f_{min}(A^*) = 2$.

Appendix D Basic notions in optimization and approximation algorithms

We give a small overview over notions in optimization and approximation algorithms needed for understanding this report. This is not a sufficient introduction to the fields in general, and only gives enough definitions to make the report self-sufficient.

Definition 3 (Linear Program)**.** *A* linear program *consists of variables x_i, linear constraints of the form $\sum_i a_{ij} \cdot x_i \geq b_j$, and a linear objective function* $\max / \min \sum_i c_i x_i$*.*

We can consider the matrix A of constraints, where each row is one constraint and each column is a variable, and obtain the form $\max / \min cx$ *s.t.* $Ax \geq b$*.*

The optimal assignment to the variables that satisfies all constraints can be found in polynomial time, for example using the Ellipsoid method.

Definition 4 (Separation oracle)**.** *A* separation oracle *is an algorithm that in time polynomial in the number of LP variables, given an assignment of variables, affirms that all constraints are satisfied if this is the case, and outputs a violated constraint otherwise.*

If an LP has a separation oracle, then the Ellipsoid algorithm can solve the LP in polynomial time, independently of the number of constraints. For instance, we could have exponentially many constraints (or even an infinite number).

Definition 5 (Dual LP)**.** *Consider a linear program* $\max cx$ *s.t.* $Ax \leq b$*, where A is the matrix of constraints. Then, the program* $\min b^T y$ *s.t.* $A^T y \geq c$ *is called the* dual *of the LP.*

Weak LP Duality states that the optimal dual value is an upper bound to the optimal primal value. *Strong LP Duality* states that if both LPs are bounded and have a solution, then their optimal values coincide.

Definition 6 (LP relaxation)**.** *Many combinatorial optimization problems can be formulated as* integer linear programs*, where we are able to restrict variables to integer values, like $x_i \in \{0, 1\}$. When* relaxing *the constraint $x_i \in \{0, 1\}$ to $0 \leq x_i \leq 1$, the resulting LP is called a* relaxation*.*

Therefore, given an integral maximization problem, the dual LP of its LP relaxation is a natural upper bound to the optimal solution.

Definition 7 (Approximation algorithm)**.** *For an optimization problem (assume without loss of generality that it is a maximization problem), let $OPT(\mathcal{I})$ denote the optimal value to instance $\mathcal{I}$. An algorithm that runs in polynomial time and for each instance $\mathcal{I}$, produces a solution of value at least $\alpha \cdot OPT(\mathcal{I})$, for some $\alpha \in [0, 1]$, is an* α-approximation*.*

Definition 8 (PTAS)**.** *A* polynomial-time approximation scheme *(PTAS) is a collection of polynomial-time algorithms A_ε for every $\varepsilon > 0$, such that A_ε is a $(1 \pm \varepsilon)$-approximation. Notice that the running time of A_ε can depend arbitrarily on ε. If the running time is polynomial in ε, we speak of a* fully polynomial-time approximation scheme *(FPTAS).*

Definition 9 (Integrality gap)**.** *Consider an LP relaxation to a maximization integer program. For an instance $\mathcal{I}$, let $OPT(\mathcal{I})$ be the optimal integral solution and $T(\mathcal{I})$ be the optimal solution to the LP relaxation. The* integrality gap *of this LP relaxation is defined as* $\sup_{\mathcal{I}} T(\mathcal{I}) / OPT(\mathcal{I})$*. For minimization problems, the integrality gap is defined analogously, but we invert the sum such that the gap is always ≥ 1.*