Submodular Max-Min Allocation under Identical Valuations

Kimon Boehmer
DIENS, École normale supérieure, PSL University, Paris, France
LIP, ENS Lyon, France

Abstract

In the problem of Submodular Max-Min Allocation, we are given a set of items, a set of players, and monotone submodular valuation functions that represent the satisfaction of a player with a certain subset of items. The goal is to find an allocation of the items to the players that maximizes the lowest satisfaction among all players.

We study this problem in the special case where all players have the same valuation function. We devise a greedy algorithm which gives a 0.4-approximation, improving the previously best factor of $\frac{10}{27} \approx 0.37$ by Uziahu and Feige.

Furthermore, we study the integrality gap of the *configuration LP* when players have identical valuations. By constructing a variable assignment to the dual from a primal integral solution, we give the first constant upper bound on the integrality gap for submodular valuations. Generalizing the result to the case where players' allocations must be independent in k given matroids, we derive a $\mathcal{O}(k)$-estimation algorithm for max-min allocation subject to k matroid constraints under identical valuations.

2012 ACM Subject Classification Theory of computation → Submodular optimization and polymatroids; Theory of computation → Linear programming; Theory of computation → Packing and covering problems

Keywords and phrases Submodularity, Approximation algorithms, Allocation, Configuration LP

Acknowledgements I want to thank Chien-Chung Huang for introducing me to the topic and for many helpful dicussions. Also, I thank the anonymous reviewers for their valuable suggestions.

1 General context

Item allocation is a fundamental problem in combinatorial optimization. Its task consists of distributing a set of *items* to 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 well-studied 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 focuses on the latter objective function. Applications of max-min allocation range from assigning classrooms to charter schools [19] to sensor placement and scheduling [18].

Usually, research focuses 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 with a certain weight, and the satisfaction of a 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 restrict our attention to the case where all players share the same submodular valuation function. It is interesting to note that the strongest hardness results of approximating submodular max-min allocation to a factor of $1 - \frac{1}{e}$ already hold even for identical valuations [17, 21].

1.1 Our results

We design a simple greedy algorithm, which we call *Truncated Max-Sum Greedy*, that achieves a 0.4-approximation for identical submodular valuations, improving the previously best-known factor of $\frac{10}{27} \approx 0.37$ by Uziahu and Feige [28]. Essentially, given a fixed threshold α, our algorithm greedily builds an allocation while ignoring players who already have items of value $\alpha \cdot \mathsf{OPT}$. Our main result is the following:

▶ **Theorem 1.1.** *There is a $\frac{2}{5}$-approximation for submodular max-min allocation under identical valuations.*

While showing a guarantee of $\frac{1}{3}$ is rather straightforward, the improvement to $\frac{2}{5}$ uses some graph-theoretic tools such as Hall's theorem and certain arguments about the digraph of reallocations needed to transform the greedy solution into an optimal solution. Our algorithm also provides guarantees in more general settings. As an example, we consider the problem of submodular max-min allocation under identical valuations with an additional *global cardinality constraint*: The objective remains the same, but we are allowed to use at most k items in total. We show the following:

▶ **Proposition 1.2.** *There is a $\frac{1}{3}$-approximation for submodular max-min allocation under identical valuations with a global cardinality constraint.*

Furthermore, we study the integrality gap of the *configuration LP* for max-min allocation, an important LP relaxation for max-min allocation.

We bound the integrality gap of the configuration LP by transforming certain primal integral solutions of the configuration LP to satisfying assignments of a dual formulation of this LP. Unlike to similar approaches when rounding configuration LPs [25], our analysis only uses the fact that the primal integral solution is *at most* OPT. As a result, our proof does not provide an algorithm (not even a non-trivial exponential one) for constructing an integral solution that demonstrates the integrality gap.

Using this technique, we give the first constant upper bound on the integrality gap of the configuration LP for identical submodular valuations.

▶ **Theorem 1.3.** *The integrality gap of the configuration LP with identical submodular valuations is at most* 3*.*

Additionally, we give a lower bound on this integrality gap.

▶ **Proposition 1.4.** *The integrality gap of the configuration LP with identical submodular valuations is at least* $4/3$*, even with* 6 *items and* 3 *players.*

We believe that the technique of Theorem 1.3 can be used to show constant integrality gaps for submodular max-min allocation in more general settings. As an example, we also consider the case where we are given k matroids on the items and the sets that players receive must all be independent in all matroids. For this case, we also give a constant upper bound on the integrality gap, independent of k.

▶ **Theorem 1.5.** *The integrality gap of the configuration LP with identical submodular valuations and any number of matroid constraints is at most* 5*.*

Accounting for the loss incurred when solving the separation oracle of the dual LP, this yields a $\mathcal{O}(k)$-estimation algorithm for max-min allocation subject to k matroid constraints.

1.2 Related work

Additive valuations. Additivity is one of the simplest assumptions on the valuation functions of the players. In the general version where players may have non-identical additive valuation functions, the state-of-the-art algorithm by Chakrabarty, Chuzhoy and Khanna [7] achieves an approximation factor of $\mathcal{O}(n^{\varepsilon})$. Interestingly, the best-known lower bound, shown by Bezáková and Dani [6], only rules out a factor better than 2.

A well-studied special case is the so-called *restricted assignment* case, where players have identical valuations, but a bipartite graph between items and players restricts the allowed allocations. To tackle this problem, Bansal and Sviridenko [4] introduced the configuration LP, and subsequent work [4, 2, 22, 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 for restricted assignment was shown to be at most 3.53 [15]. For identical valuations, Woeginger designed a polynomial-time approximation scheme (PTAS) [30].

Submodular valuations. The first approximation algorithm for general submodular valuations was given in [14], and even though improvements have been made [13], the best-known approximation factor remains polynomial in the input. For a constant number of players, Chekuri, Vondrák and Zenklusen [8] gave a $1 - 1/e$-approximation.

For max-sum, Vondrak gave a $1 - 1/e$-approximation [29], which was shown to be optimal [17, 21]. In the restricted assignment case, Bamas, Morell and Rohwedder [3] generalized the techniques from Bansal and Sviridenko to obtain a $\mathcal{O}(\log \log n)$-approximation. Under identical valuations, Krause, Rajagopal, Gupta and Guestrin [18] 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.

More recently, another line of research has studied the problem of finding an allocation which approximates the *maximin share* of all players, which is the largest value a player can ensure by proposing an allocation and receiving the bundle which is worst for them. Under identical valuations, this problem is equivalent to submodular max-min allocation. Barman and Krishnamurthy [5] gave a different algorithm with the same guarantee as Krause et al [18]. Ghodsi, Hajiaghayi, Seddighin, Seddighin and Yami [12] improved this factor to $\frac{1}{3}$. Finally, Uziahu and Feige [28] gave a $\frac{10}{27}$-approximation by analyzing a bidding procedure which uses a truncated version of the valuation function.

Machine scheduling. If we reverse the roles of *max* and *min*, we obtain the classical *makespan minimization* setting, where we try to assign jobs to machines so as to minimize the latest completion time. As its max-min counterpart, it admits a PTAS for identical additive valuations [16]. The submodular variant of makespan minimization, sometimes called *submodular load balancing*, is not $o\left(\sqrt{\frac{n}{\log n}}\right)$-approximable [27]. This stands in stark contrast to its "dual" max-min variant.

2 Preliminaries

Consider a set J (with $n := |J|$) of items and a *valuation function* $f : 2^J \to \mathbb{R}$ 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 the case where f is *submodular*: A function $f : 2^J \to \mathbb{R}$ 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 so a monotone function is non-negative. In Appendix A, we give some basic facts about submodular functions that will be used implicitly throughout our proofs.

We define P (with $m := |P|$) to be the set of *players*. Since we are focusing on identical valuations, each player p has the same valuation function f describing their satisfaction with each 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 $j \in \bigcup_{p \in P} A_p$. If every item is allocated, we speak of an *allocation*. We refer to the sets of items in a (partial) allocation as *configurations*. We define $f_{\mathsf{sum}}(A) = \sum_{p \in P} f(A_p)$ and $f_{\mathsf{min}}(A) := \min_{p \in P} f(A_p)$. We will refer to players p with $f(A_p) = f_{\mathsf{min}}(A)$ as *min players*.

We briefly define matroids since we discuss them in Section 4.3. A *matroid* $\mathcal{M} = (J, \mathcal{I})$ is a non-empty subset-closed set system in which for any $J' \subseteq J$, all sets of $\mathcal{I}$ restricted to J' which are maximal have the same cardinality. The sets in $\mathcal{I}$ are called *independent*.

2.1 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 $f : 2^J \to \mathbb{R}$.
Output: An allocation A that maximizes $f_{\mathsf{min}}(A)$

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 $f(S)$.

Approximation Hardness. Khot, Lipton, Markakis and Mehta [17] showed that when players have identical valuations, it is **NP**-hard to distinguish between the case where all players can achieve value 1 and the case where a player achieves at most value $1 - \frac{1}{e} + \varepsilon$ on average. Since the average value is always at least as big as the min value, their hardness result, while originally designed for max-sum, immediately translates to our setting:

▶ **Corollary 2.1** ([17]). *Unless* $\mathbf{P} = \mathbf{NP}$, Submodular Max-Min Allocation *cannot be approximated within a factor* $1 - \frac{1}{e} + \varepsilon$ *for any* $\varepsilon > 0$.

3 Truncated Max-Sum Greedy

In this section, we present our greedy algorithm for Submodular Max-Min Allocation.

Fix a threshold $\alpha \leq 1$. We assume that the algorithm knows the optimal value OPT. [1]

[1] We assume that the size of a rational representation of $f(S)$ is at most polynomial in $n + m$. We can therefore scale f such that all values are integral and perform a binary search with start interval $[0, \overline{f}(J)]$, where $\overline{f}$ is the scaled version of f. Under our assumption, the binary search procedure will only recurse a polynomial number of times.

First, the algorithm takes the m items j with largest $f(j)$-value and allocates one of them to each player. Then, it greedily chooses the pair of available item j and player p that maximizes the marginal contribution $\Delta_f(j \mid A_p)$, under the constraint that $f(A_p) < \alpha \cdot \mathsf{OPT}$, i.e. we only consider players that have not reached an α-fraction of their desired value.

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 $f : 2^J \to \mathbb{R}$
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* $f(A_p) < \alpha \cdot \mathsf{OPT}$ **do**
  $(j^*, p^*) \leftarrow \arg\max_{j \in J, p \in P : f(A_p) < \alpha \cdot \mathsf{OPT}} \Delta_f(j \mid A_p)$;
  $J \leftarrow J - \{j^*\}$;
  $A_{p^*} \leftarrow A_{p^*} \cup \{j^*\}$;
end
return $(A_1, ..., A_m)$

▶ **Observation 3.1.** *We can assume* $f(j) < \alpha \cdot \mathsf{OPT}$ *for all* $j \in J$.

Proof. We show that if the truncated max-sum greedy gives an α-approximation on instances where for all items $j \in J$, $f(j) < \alpha \cdot \mathsf{OPT}$, then it also gives an α-approximation for the general case.

Let OPT denote the optimal value in the general instance. Assume that there are k *big* items j with $f(j) \geq \alpha \cdot \mathsf{OPT}$. In an optimal allocation, these items can be given to at most k players. Thus, there exists an instance with $\max(0, m-k)$ players and a subset $J' \subseteq \{j \in J \mid f(j) < \alpha \cdot \mathsf{OPT}\}$ of items that achieve at least OPT.

If greedy is performed on the instance with $\max(0, m-k)$ players and without the big items, then it will obtain a solution of value at least $\alpha \cdot \mathsf{OPT}$ (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 $\alpha \cdot \mathsf{OPT}$ to the general instance. ◀

As a warm-up, we first show that we obtain a $\frac{1}{3}$-approximation when setting $\alpha = \frac{1}{3}$. Afterwards, we will show how to improve the factor to $\frac{2}{5}$ using a more refined analysis.

Consider a greedy allocation A and an optimal allocation A^*. Let q be a min player in A, i.e. $f(A_q) = f_{\mathsf{min}}(A)$. We may assume $n \geq m$, since otherwise the optimal value is 0. We first show the following simple but crucial lemma.

▶ **Lemma 3.2.** *Assume* $f(A_q) < \alpha \cdot \mathsf{OPT}$. *Then,*

$$\sum_{j \in A^*} \Delta_f(j \mid A_q) \leq \sum_{p \in P - \{q\}} f(A_p)$$

Proof. For any item j, let $A^{(j)}$ denote the partial allocation of the algorithm immediately before j is allocated. We have $\sum_{j \in A_q} \Delta_f(j \mid A_q) = 0$. For all other $p \in P - \{q\}$, we have

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

The first inequality follows from the submodularity of f, since $A_q^{(j)} \subseteq A_q$. 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 $f(A_p) < \alpha \cdot \mathsf{OPT}$. Therefore,

$$\sum_{j \in A^*} \Delta_f(j \mid A_q) = \sum_{p \in P-\{q\}} \sum_{j \in A_p} \Delta_f(j \mid A_q) + \sum_{j \in A_q} \Delta_f(j \mid A_q) \leq \sum_{p \in P-\{q\}} f(A_p). \tag{2}$$

◀

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

▶ **Lemma 3.3.** *For all players $p \in P$, $f(A_p) \leq 2 \cdot \alpha \cdot \mathsf{OPT}$.*

Proof. Player p can only be allocated one item j that surpasses $\alpha \cdot \mathsf{OPT}$. This item has size $\leq \alpha \cdot \mathsf{OPT}$, by Observation 3.1. Let A' be the partial allocation before j is allocated to p. Then,

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

◀

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

▶ **Lemma 3.4.** $f_{\mathsf{min}}(A^*) < f(A_q) + \frac{1}{m} \sum_{p \in P} f(A_p)$.

Proof.

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

(3) uses monotonicity of f, (4) uses Lemma 3.2 and (5) holds since $f(A_q) > 0$ by our assumption $n \geq m$ and the fact that greedy starts by allocating one item to each player. ◀

To see that when setting $\alpha = \frac{1}{3}$, this indeed implies the approximation ratio of $\frac{1}{3}$, substitute $\frac{1}{m} \sum_{p \in P} f(A_p) \leq 2 \cdot \alpha \cdot \mathsf{OPT}$ by Lemma 3.3. Taking $2 \cdot \alpha \cdot \mathsf{OPT} = \frac{2}{3} \cdot \mathsf{OPT}$ on the other side, we obtain $\frac{1}{3}\mathsf{OPT} < f(A_q)$.

▶ Remark 3.5. This algorithm extends to the case where we are additionally given a *global cardinality constraint*. That is, we must choose a subset of at most k items to allocate.

To see this, consider again an optimal allocation A^* and the greedy allocation A. Notice that in our new setting, we might have $\bigcup_{p \in P} A_p \neq \bigcup_{p \in P} A_p^*$ and therefore the first equality

in (2) no longer holds. However, we can circumvent this issue as follows. We can assume that both A and A^* contain exactly k items by monotonicity. Let $b : \bigcup_{p\in P} A^*_p \to \bigcup_{p\in P} A_p$ be any bijection such that $b(j) = j$ for all $j \in (\bigcup_{p\in P} A^*_p) \cap (\bigcup_{p\in P} A_p)$. For all $j \in A$, let $A^{(j)}$ be the partial greedy allocation exactly before allocating j. Consider any $j \in A^*$ and let $b(j) \in A_p$. We have

$$\Delta_f(j \mid A_q) \leq \Delta_f(j \mid A_q^{(b(j))}) \leq \Delta_f(b(j) \mid A_p^{(b(j))}) \tag{6}$$

using that when $b(j)$ is allocated, item j is still available and thus greedy must prefer allocating $b(j)$ to p than allocating j to q. By (6) and the fact that b is a bijection,

$$\begin{aligned}\sum_{j\in A^*} \Delta_f(j \mid A_q) &\leq \sum_{p\in P-\{q\}} \sum_{j\in A^*:b(j)\in A_p} \Delta_f(b(j) \mid A_p^{(b(j))}) \\ &= \sum_{p\in P-\{q\}} \sum_{j\in A_p} \Delta_f(j \mid A_p^{(j)}) \\ &= \sum_{p\in P-\{q\}} f(A_p)\end{aligned}$$

reproving Lemma 3.2 in this more general setting. The rest of the analysis can be done as for the unconstrained case. Therefore, Proposition 1.2 holds.

3.1 Ratio of 2/5

We now prove that if $\alpha = \frac{2}{5}$, then $f_{\mathsf{min}}(A) \geq \frac{2}{5}\mathsf{OPT}$. To this end, suppose for contradiction that there exists a counterexample to this claim. We assume that this instance (P, J, f) is minimal with respect to $|P|$ and that f is scaled in such a way that $f_{\mathsf{min}}(A^*) = \mathsf{OPT} = 5$. We also fix a min player q with $f(A_q) \leq 2$. Note that we could have assumed $f(A_q) < 2$ but this is not necessary for the proof.

Let us define $\mathsf{avg} := \frac{1}{m}\sum_{p\in P} f(A_p)$. From Lemma 3.4 we obtain $\mathsf{avg} > 3$. Our goal is to show that $\mathsf{avg} \leq 3$, thereby arriving at a contradiction.

Letting a_p^i denote the i-th item allocated to p, define $\beta := \max_{p\in P:|A_p|\geq 2} \Delta_f(a_p^2 \mid \{a_p^1\})$ as the largest marginal contribution of an item during the greedy algorithm which is not the first item of a player. Notice that it suffices to consider a_p^2, since the marginal contributions of later items of a player can only be smaller. 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 3.1. Notice finally that for $p \in P$, we have $f(a_p^1) \geq \beta$, as otherwise the maximizer a_p^2 in the definition of β would have been among the largest m items, and could not be the second item of a player.

▶ **Lemma 3.6.** *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 and all items in $\bigcup_{i=1}^k A_{p_i}$ from the instance 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 $f_{\mathsf{min}}(A') \leq 2$. On the other hand, since all removed items only belonged to k players in the optimal solution, removing these k players gives a solution A^{**} with $f(A^{**}_p) \geq 5$ for all remaining players p. ◀

The above lemma allows us to prove the existence of a "fixed" item for each player:

▶ **Lemma 3.7.** *There is a permutation $\phi : P \to P$ such that $A_p \cap A^*_{\phi(p)} \neq \emptyset$ for all $p \in P$.*

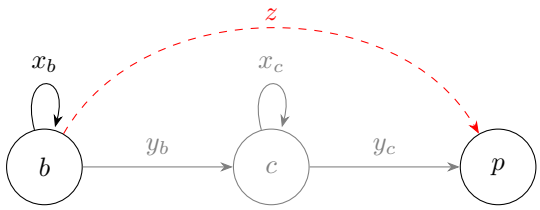

Figure 1 Visualization of Lemma 3.9.

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 $A_p \cap A^*_{p'} \neq \emptyset$. Let $N(Q) = \bigcup_{q \in Q}\{v \in V \mid (q, v) \in E\}$ be the neighbourhood of Q. Consider an arbitrary non-empty subset $Q \subseteq P - \{q\}$. By Lemma 3.6, $|N(Q)| \geq |Q| + 1$. For the set $\{q\}$, we have $|N(\{q\})| \geq 1$ since $A_q \neq \emptyset$ and each item in q must be allocated somewhere in the optimal solution. For the set $\emptyset$, we have $|N(\emptyset)| \geq 0 = |\emptyset|$. For any other set $\{q\} \subset Q \subseteq P$, we know that $|N(Q)| \geq |N(Q - \{q\})| \geq |Q - \{q\}| + 1 = |Q|$. Altogether, 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 defines exactly the desired permutation ϕ. ◀

To simplify notation, we consider an optimal solution such that ϕ is the identity function, by just permuting the allocations of the players in the optimal solution. We now define the *reallocation digraph* $G = (P, E)$: Each pair of players $(p, r) \in P^2$ is associated to a set $A_{p\to r} = A_p \cap A^*_r$. An arc (p, r) exists if $A_{p\to r} \neq \emptyset$. By Lemma 3.7, each vertex has a self-loop.

We subdivide the players of our instance into two categories $P = B \cup S$. The set B of *big* players consists of all players b such that $f(A_b) > 4 - \beta$. 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 3.8.** *For any $p \in B$, it holds that $|A_p| = 2$.*

Proof. Any player $p \in B$ cannot be allocated just one item j since $f(j) < 2 \leq 4 - \beta \leq f(A_p)$. If p was allocated k items for $k > 2$, then $f(\{a^1_p, \ldots, a^{k-1}_p\}) < 2$ must hold in order for the greedy algorithm to allocate a^k_p to p. But since $f(a^1_p) \geq \beta$, we must have $\Delta_f(a^k_p \mid \{a^1_p, \ldots, a^{k-1}_p\}) \leq \Delta_f(a^2_p \mid \{a^1_p\}) \leq 2 - \beta$. So we have

$$f(A_p) = f(A_p - a^k_p) + \Delta_f(a^k_p \mid \{a^1_p, \ldots, a^{k-1}_p\}) < 2 + 2 - \beta \leq f(A_p),$$

contradicting our assumption $k > 2$. ◀

For each $p \in B$, denote by x_p the item in $A_{p\to p}$ and denote by y_p the other item of A_p. Lemma 3.7 guarantees the existence of the first item and Observation 3.8 implies that there is exactly one other item. Lemma 3.6 guarantees that the other item is not in $A_{p\to p}$.

▶ **Lemma 3.9.** *For any $b, c \in B$ with $b \neq c$, $(b, c) \notin E$.*

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 $y_c \in A^*_p$. Consider a modified instance where we remove player c as well as items $\{y_b, x_c, y_c\}$ and add an item z which contributes an additive factor of $t := \max(f(y_b), \Delta_f(y_c \mid A^*_p - \{y_c\}))$ to every set. By Observation A.2 and Lemma A.1, this function is submodular and monotone. Notice that this addition of z does not change any marginal contribution, i.e. for $j \neq z$ we have $\Delta_f(j \mid S) = \Delta_f(j \mid S \cup \{z\})$.

The greedy algorithm on the new instance may allocate all items that give marginal contribution larger than t in the original run as before. 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, and greedy considers b as $f(A_b) = f(\emptyset) < 2$ or $f(A_b) = f(x_b) < 2$).

Since $b \in B$, we have $f(A_b) > 4 - \beta \geq 2$. Furthermore, $f(\{x_b, z\}) \geq f(x_b) + f(y_b) \geq f(A_b) \geq 2$. Consider the moment in the greedy algorithm after z was allocated to b. If x_b had already been allocated, then b has reached the threshold of 2 and will be thus ignored by the greedy algorithm. Therefore, all following items can be allocated to the same players as in the original run of the greedy algorithm. Now suppose x_b had not yet been allocated. Since the addition of z to player b does not influence the marginal contributions and $f(z) < 2$, the greedy algorithm can allocate all items up to x_b in the same way (including x_b). After allocating x_b, player b has reached the threshold of 2 and thus again the greedy algorithm can allocate in the same way as in the original run. Thus, we again obtain $f_{\mathsf{min}}(A'_q) \leq 2$, 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

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

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

Thus, in fact, $G[B]$ is an independent set. We now argue that B cannot be too large. To this end, we first prove the following two helpful lemmata. 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 3.10.** *For any player p, $\sum_{j\in W_p} \Delta_f(j \mid A_q) < 2$.*

Proof. Recall that $A^{(j)}$ denotes the partial greedy allocation exactly before item j is allocated. Let ℓ be the last item allocated to A_p. By Lemma 3.7, there exists an item $k \in A_{p\to p}$. Therefore, $W_p \subseteq A_p - \{k\}$ and we can write

$$\begin{aligned} \sum_{j\in W_p} \Delta_f(j \mid A_q) &\leq \sum_{j\in A_p - \{k\}} \Delta_f(j \mid A_q) \\ &\leq \sum_{j\in A_p - \{k\}} \Delta_f(j \mid A_q^{(j)}) \\ &\leq \Delta_f(\ell \mid A_p^{(\ell)}) + \sum_{j\in A_p - \{k,\ell\}} \Delta_f(j \mid A_p^{(j)}) \\ &\leq \Delta_f(\ell \mid A_p^{(k)}) + \sum_{j\in A_p - \{k,\ell\}} \Delta_f(j \mid A_p^{(j)}) \\ &\leq \Delta_f(k \mid A_p^{(k)}) + \sum_{j\in A_p - \{k,\ell\}} \Delta_f(j \mid A_p^{(j)}) \\ &= f(A_p - \{\ell\}) < 2 \end{aligned}$$

The third inequality is because the items were allocated to p and not q. The fourth inequality comes from $A_p^{(\ell)} \supseteq A_p^{(k)}$, then we use that at the partial allocation $A^{(k)}$, greedy prefered

k over ℓ. Finally, the strict inequality holds because otherwise p would have achieved the $\alpha \cdot \mathsf{OPT}$ threshold and not received any more items. ◀

▶ **Lemma 3.11.** *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. First, notice that

$$\sum_{j \in A_p^*} \Delta_f(j \mid A_q) \geq \Delta_f(A_p^* \mid A_q) \geq f(A_p^*) - f(A_q) \geq 5 - 2 = 3. \tag{7}$$

Since $V_p = A_p^* - (A_p - W_p)$, $A_p - W_p \subseteq A_p^*$, and $W_p \subseteq A_p$, we can write

$$\sum_{j \in V_p} \Delta_f(j \mid A_q) = \sum_{j \in A_p^*} \Delta_f(j \mid A_q) + \sum_{j \in W_p} \Delta_f(j \mid A_q) - \sum_{j \in A_p} \Delta_f(j \mid A_q)$$

Now, the result follows by using (7) and (1). ◀

▶ **Lemma 3.12.** $|B| \leq |S|$.

Proof. 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. If we can show $S_{\geq 2} = \emptyset$, this means that no two big players have an edge towards the same small player. But since every big player has an edge towards a small player by Lemma 3.9, by pigeonhole principle this would imply the desired $|B| \leq |S|$.

Associate with each player p the quantity

$$g(p) := \sum_{j \in W_p} \Delta_f(j \mid A_q) - \sum_{j \in V_p} \Delta_f(j \mid A_q)$$

Note that $\sum_{p \in P} g(p) = 0$, as $\bigcup_{p \in P} W_p = \bigcup_{p \in P} V_p$. We can decompose the set of players as

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

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

$$\sum_{p \in P} g(p) = \sum_{p \in S_0} g(p) + \sum_{s \in S_1} g(s) + g(b_1(s)) + \sum_{s \in S_{\geq 2}} g(s) + g(b_1(s)) + ... + g(b_k(s)) \tag{8}$$

Since $\sum_{p \in P} g(p) = 0$, to prove $S_{\geq 2} = \emptyset$, it is enough to show that all terms of (8) are ≤ 0, and that terms of the last sum are < 0. This is the purpose of the rest of the proof.

1. $g(p) \leq 0$ for $p \in S_0$. This follows directly from Lemma 3.11 and $f(A_p) \leq 4 - \beta \leq 3$.
2. $g(s) + g(b) \leq 0$ for $s \in S_1$ and $b = b_1(s)$. Since s is a small player, we have $f(A_s) \leq 4 - \beta$, and by Lemma 3.11, we get $g(s) \leq 1 - \beta$. If we can show $g(b) \leq \beta - 1$, then this implies $g(s) + g(b) \leq 0$. Recall that $W_b = \{y_b\}$.
   As the second item allocated to b contributes at most β, we have $\Delta_f(x_b \mid A_q) + \Delta_f(y_b \mid A_q) \leq 2 + \beta$. By the facts $f(A_b^*) - f(A_q) \geq 3$ and $A_b^* = V_b \cup (A_b - W_b)$, we get

$$\begin{aligned}
\sum_{j \in V_b} \Delta_f(j \mid A_q) &\geq \Delta_f(V_b \cup (A_b - W_b) \mid (A_b - W_b) \cup A_q) \\
&= \Delta_f(A_b^* \mid (A_b - W_b) \cup A_q) \\
&\geq f(A_b^*) - \Delta_f(A_b - W_b \mid A_q) - f(A_q) \\
&\geq 3 - \Delta_f(x_b \mid A_q)
\end{aligned}$$

   In total, $g(b) \leq \Delta_f(y_b \mid A_q) - \sum_{j \in V_b} \Delta_f(j \mid A_q) \leq \Delta_f(y_b \mid A_q) - 3 + \Delta_f(x_b \mid A_q) \leq \beta - 1$.

3. $g(s)+g(b)+g(c)<0$ for $s\in S_{\geq 2}$, $b=b_1(s)$ and $c=b_2(s)$. 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

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

The same holds for player c. By Lemma 3.10, we also know $g(s)<2-\Delta_f(y_b\mid A_q)-\Delta_f(y_c\mid A_q)$. Therefore

$$g(b)+g(c)+g(s)\leq \Delta_f(y_b\mid A_q)-1+\Delta_f(y_c\mid A_q)-1+g(s)<0.$$

◀

We can conclude as follows:

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

where the last inequality is because $2+\beta\geq 4-\beta$, and $|B|\leq|S|$ by Lemma 3.12. This contradicts $\mathsf{avg}>3$, as stated in the beginning. This finishes the proof of Theorem 1.1.

We discuss the (close-to-) tightness of this analysis in Appendix B. In particular, we show that no value of α can lead to an approximation factor better than $\frac{2+S}{4+3S}\approx 0.4068$, where $S=\sum_{i=1}^{\infty}\frac{1}{s_i-1}$ with $s_k=1+\prod_{i=1}^{n}s_i$ is a constant related to the *Sylvester sequence* or *Salzer sequence* and was already used in [20, 24] for other packing problems.

4 Integrality gaps for the configuration LP

Assignment LP (A)

$$\sum_{i=1}^{m} x_{ij}=1 \qquad \forall j=1,\dots,n$$

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

$$x_{ij}\geq 0 \qquad \forall i,j$$

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 2 The assignment LP and the configuration LP.

In this section, we study the integrality gap of the *configuration LP* in the case of identical valuation functions. Before defining this LP, let us consider the simpler *assignment LP*: It searches for the largest T such that the program depicted in figure 2 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 identical additive valuations.

Dual LP (D0)

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

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 3 The duals to the configuration LP.

In the restricted assignment case, (A) has a bad integrality gap [4]. 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 2 on the right. The largest T such that this LP has a feasible solution is an upper bound on the optimal solution. We call this system (P).

It is well-known that the integrality gap of (P) in the general case is $\Omega\big(\sqrt{m}\big)$ [4], while in the restricted assignment case it was shown to be smaller than 3.534 [15]. When valuation functions are submodular, the gap in the restricted assignment case is $\mathcal{O}(\log \log n)$ [3].

(P) has exponentially many variables in general. Assuming that the primal minimizes the 0-vector, we can formulate its dual (D0), depicted in figure 3 on the left. The separation problem for this LP is the classical Knapsack problem. Since Knapsack admits a (F)PTAS [23], we can solve the LP up to any desired accuracy using the Ellipsoid method.

4.1 Constructing Dual Solutions

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 arbitrarily 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 3.

In the case of identical valuations, we can, without loss of generality, set all player variables to 1, and obtain (D2), as seen on the right of figure 3. This LP searches for a fractional hitting set of size smaller than m that hits all configurations larger than our threshold value. In (D2), we call the first constraint *value constraint* and the second set of constraints *configuration constraints*. If (D2) has a feasible solution, then (D0) is unbounded, and by weak duality, (P) is infeasible. We obtain the following:

▶ **Proposition 4.1.** *Consider a class Γ of max-min allocation instances, and let $\mathsf{OPT}(\mathcal{I})$ be the optimal integral value of instance $\mathcal{I}$. If for all $\mathcal{I} \in \Gamma$, (D2) is feasible for $\mathcal{I}$ and threshold value $\alpha \cdot \mathsf{OPT}(\mathcal{I})$, then the integrality gap of the configuration LP on Γ is at most α.*

We give one more useful observation:

▶ **Observation 4.2.** *If there are variables y_j such that the configuration constraints of (D2) are satisfied, $\sum_{j\in J} y_j \leq m$ and there exists $j^* \in J$ such that for all $C \ni j^*$, we have $\sum_{j\in C} y_j > 1$, then (D2) is feasible.*

Proof. Since y_{j^*} is not involved in any tight constraint, we can lower its value by a sufficiently small amount, satisfying the value constraint while keeping all configuration constraints satisfied. ◀

4.2 Submodular valuations

The system (P) can also be formulated for submodular valuations. While in the additive case, solving the separation problem only incurred an ε error, this is different in the submodular case. This is because we need to solve a submodular version of knapsack for the separation oracle. Due to the hardness of maximum coverage [11], which includes submodular knapsack, we lose a factor of $1 - \frac{1}{e}$ while solving the LP. On the algorithmic side, this hardness is matched by a $1 - \frac{1}{e}$-approximation by Sviridenko [26]. Therefore, any rounding algorithm that loses a factor of α translates into an approximation algorithm with factor $(1 - \frac{1}{e})\alpha$.

In the following, we show that apart from our combinatorial approach presented in the last section, an LP rounding approach could also lead to a constant-factor approximation..

▶ **Theorem 1.3.** *The integrality gap of the configuration LP with identical submodular valuations is at most* 3*.*

We first show that we can restrict our attention to items j with $f(j) \leq \mathsf{OPT}$. Let $\mathsf{OPT}(\mathcal{I})$ denote the optimal integral value for an instance $\mathcal{I}$, and let $\mathsf{OPT}_{\mathsf{LP}}(\mathcal{I})$ denote the optimal value of (P) on $\mathcal{I}$.

▶ **Lemma 4.3.** *Let $\mathcal{I}$ be an instance with an item j such that $f(j) > \mathsf{OPT}(\mathcal{I})$. Let $\mathcal{I}'$ be the instance obtained from $\mathcal{I}$ by removing j and one player. Then, $\mathsf{OPT}(\mathcal{I}) \geq \mathsf{OPT}(\mathcal{I}')$ and $\mathsf{OPT}_{\mathsf{LP}}(\mathcal{I}) \leq \mathsf{OPT}_{\mathsf{LP}}(\mathcal{I}')$.*

Proof. Fix an arbitrary player $p \in P$. For an optimal solution x to (P), we have

$$\sum_{C \ni j} \sum_{p \in P} x_{p,C} \leq 1 \leq \sum_{C \in \mathcal{C}(\mathsf{OPT}_{\mathsf{LP}}(\mathcal{I}),p)} x_{p,C}$$

Therefore, we can replace all configurations in which j appears by the conifigurations of player p, without violating any constraint except the constraint of player p. Since item j is not used anymore, this is a valid solution to (P) on instance $\mathcal{I}'$ of value $\mathsf{OPT}_{\mathsf{LP}}(\mathcal{I})$.

Now let A be an optimal integral allocation of $\mathcal{I}'$ and assume for contradiction $f_{\mathsf{min}}(A) > \mathsf{OPT}(\mathcal{I})$. Adding a new player r with $A_r = \{j\}$ gives an integral allocation for $\mathcal{I}$ of value $\min(f_{\min}(A), f(j)) > \mathsf{OPT}(\mathcal{I})$, a contradiction. ◀

Therefore, if our instance contains items that contribute more than OPT, then we can instead consider a smaller instance with larger integrality gap. We now prove Theorem 1.3.

Proof. Consider an arbitrary instance $\mathcal{I}$ of our problem and scale the valuation function so that $\mathsf{OPT}(\mathcal{I}) = 1$. We can assume that for all $j \in J$, $f(j) \leq 1$ by Lemma 4.3. Let $\overline{f}(S) := \min(2, f(S))$, and consider an allocation A that maximizes $\sum_{p \in P} \overline{f}(A_p)$.

For $j \in A_p$ and f, define $\Delta_f(-j \mid A_p) := \Delta_f(A_p \mid A_p - \{j\})$. For any $p \in P$, let $T_f(p) := \sum_{j \in A_p} \Delta_f(-j \mid A_p)$. For each $p \in P$ and $j \in A_p$, we set the variables in (D2) to

$$y_j = \frac{\Delta_{\overline{f}}(-j \mid A_p)}{T_{\overline{f}}(p)}$$

if $T_{\overline{f}}(p) \neq 0$, and $y_j = 0$ otherwise. For notational convenience, define $y(S) = \sum_{j \in S} y_j$. We prove that y is a (strictly) feasible solution to (D2) when $T = 3$. Suppose for contradiction that there is a set C of items with $f(C) > 3$ but $y(C) \leq 1$.

Let q be the min player in A. Clearly, $f(A_q) \leq 1$, and thus $\Delta_f(C \mid A_q) > 2$. Let $j^* = \arg\max_{j \in C} \frac{\Delta_f(j \mid A_q)}{y_j}$. If $\frac{\Delta_f(j^* \mid A_q)}{y_{j^*}} \leq 2$, then we would have

$$\frac{\Delta_f(C \mid A_q)}{y(C)} \leq \frac{\sum_{j \in C} \Delta_f(j \mid A_q)}{y(C)} \leq \frac{\sum_{j \in C} 2 \cdot y_j}{y(C)} = 2$$

which contradicts our assumption $y(C) \leq 1$. Therefore, we can assume $\frac{\Delta_f(j^* \mid A_q)}{y_{j^*}} > 2$.

We want to argue that reallocating item j^* to player q increases our potential according to $\overline{f}$. Notice that the gain of adding j^* to player q is $\Delta_{\overline{f}}(j^* \mid A_q)$, while the loss of removing j^* from player p is $\Delta_{\overline{f}}(-j^* \mid A_p) = y_{j^*} \cdot T_{\overline{f}}(p)$. We now bound $T_{\overline{f}}(p)$. First, we have

$$\begin{aligned} T_{\overline{f}}(p) &= \sum_{j \in A_p} \Delta_{\overline{f}}(-j \mid A_p) \\ &= \sum_{j \in A_p} \min\Big(0, \Delta_f(-j \mid A_p) - \max\Big(0, f(A_p) - 2\Big)\Big) \\ &= \sum_{j \in A_p : \Delta_f(-j \mid A_p) \geq f(A_p) - 2} \Delta_f(-j \mid A_p) - \max\Big(0, f(A_p) - 2\Big) \end{aligned}$$

If this sum is empty, then $T_{\overline{f}}(p) = 0$. Otherwise, we can assume that $\max\big(0, f(A_p) - 2\big)$ is subtracted at least once. Thus,

$$\begin{aligned} T_{\overline{f}}(p) &\leq \Big(\sum_{j \in A_p} \Delta_f(-j \mid A_p)\Big) - \max\Big(0, f(A_p) - 2\Big) \\ &\leq f(A_p) - \max\Big(0, f(A_p) - 2\Big) \\ &\leq 2 \end{aligned}$$

where the second inequality is by submodularity of f. Therefore,

$$\Delta_{\overline{f}}(j^* \mid A_q) - \Delta_{\overline{f}}(-j^* \mid A_p) > \Delta_{\overline{f}}(j^* \mid A_q) - 2 \cdot y_{j^*} = \Delta_f(j^* \mid A_q) - 2 \cdot y_{j^*} > 0,$$

where the equality is because of $f(A_q \cup \{j^*\}) = \overline{f}(A_q \cup \{j\})$, which follows from our assumption $f(j^*) \leq 1$ and $f(A_q) \leq 1$. So, reallocating j^* from p to q increases the potential $\overline{f}$, contradicting the optimality of A.

We conclude that the configuration constraints in (D2) are satisfied and not tight for $T > 3$. We can finish the proof with Observation 4.2, since

$$\sum_{j \in J} y_j = \sum_{p \in P} \sum_{j \in A_p} y_j \leq m.$$

◀

A simple lower bound on the integrality gap is proven in Appendix C.

▶ **Proposition 1.4.** *The integrality gap of the configuration LP with identical submodular valuations is at least* $4/3$*, even with* 6 *items and* 3 *players.*

4.3 Submodular valuations with matroid constraints

Now consider the following problem: Apart from items J, players P and a submodular monotone valuation function f, we are given k matroids $\mathcal{M}_1 = (J, \mathcal{I}_1), \ldots, \mathcal{M}_k = (J, \mathcal{I}_k)$ and our task is to find a (partial) allocation A such that $A_p \in \mathcal{I}_i$ for all $p \in P$ and all $i \in [k]$. The configuration LP can be formulated in the same way as before, and the separation oracle incurs a loss of $2.7k$ by using the algorithm of [9]. In the following, we prove Theorem 1.5, which states that the integrality gap is still bounded by 5 in this case and implies a $13.5k$-estimation algorithm for max-min allocation subject to k matroid constraints.

▶ **Theorem 1.5.** *The integrality gap of the configuration LP with identical submodular valuations and any number of matroid constraints is at most* 5*.*

Proof. As before, let $\mathsf{OPT} = 1$, $\overline{f}(S) := \min(2, f(S))$ and define A and the y_j-variables in the same way. The proof of Lemma 4.3 directly translates to the matroid constraint setting, so we may again assume $f(j) \leq 1$ for all $j \in J$. Unlike before, we set $y_j = 0$ for all $j \in A_q$, where q is a fixed min player in A. We also set $y_j = 0$ for all unallocated $j \notin A$.

Now suppose for contradiction that there is a set $D \in \bigcap_{i=1}^{k} \mathcal{I}_k$ of items with $f(D) > 5$ but $y(D) < 1$. We can partition D into $D_1 \uplus D_2$ such that $f(D_i) \geq 2$ for both $i \in [2]$, by greedily adding items to D_1 until $f(D_1) > 2$. By the assumption that $f(j) \leq 1$ for all $j \in J$ and submodularity, this implies $f(D_2) = f(D - D_1) > 2$. Now, choose the partition D_i with the smaller y-value, and call it C. As a subset of D, clearly C is independent in all matroids, and $y(C) < \frac{1}{2}$.

Consider reallocating all items from C to player q and discarding all original items in A_q. We want to argue that this modified allocation has higher potential. The gain for player q is $\overline{f}(C) - \overline{f}(A_q) \geq 2 - 1 = 1$. The loss for all other players is

$$\begin{aligned}\sum_{p \in P - \{q\}} \Delta_{\overline{f}}(-C \mid A_p) &\leq \sum_{p \in P-\{q\}} \sum_{j \in A_p} \Delta_{\overline{f}}(-j \mid A_p) \\ &= \sum_{p \in P-\{q\}} \sum_{j \in A_p} y_j \cdot T_{\overline{f}}(p) \\ &\leq y(C) \cdot 2 \\ &< 1\end{aligned}$$

since we can bound $T_{\overline{f}}(p)$ as in the proof of Theorem 1.3. The total potential change is negative. This contradicts our choice of A. Hence, the configuration constraints are all satisfied. The value constraint is satisfied since

$$\sum_{j \in J} y_j = \sum_{j \notin A} y_j + \sum_{j \in A_q} y_j + \sum_{p \in P-\{q\}} \sum_{j \in A_p} y_j = \sum_{p \in P-\{q\}} \sum_{j \in A_p} y_j \leq m - 1 < m.$$

◀

Notice that we could replace matroids by any set system closed under subsets, but then of course we might not be able to solve the configuration LP efficiently.

Conclusion

We presented an improved approximation algorithm for submodular max-min allocation under identical valuations, and showed some constant upper bounds on the integrality gap of the configuration LP under identical valuations.

However, several natural questions remain open. First, the proposed 0.4-approximation is still far from the best-known complexity-theoretical hardness bound of $1 - \frac{1}{e} \approx 0.63$. Another promising direction is to further explore the configuration LP. It would be valuable to obtain constructive, polynomial-time rounding algorithms that achieve the proven integrality gaps. Furthermore, it would be interesting to see by how much we can relax the assumption of identical valuations. A natural first step in this direction would be to determine the integrality gap of the configuration LP in the case of submodular "related machines", where each player can be satisfied with a different value.

References

1 Chidambaram Annamalai, Christos Kalaitzis, and Ola Svensson. Combinatorial algorithm for restricted max-min fair allocation. *ACM Transactions on Algorithms (TALG)*, 13(3):1–28, 2017.

2 Arash Asadpour, Uriel Feige, and Amin Saberi. Santa claus meets hypergraph matchings. *ACM Transactions on Algorithms (TALG)*, 8(3):1–9, 2012.

3 Étienne Bamas, Sarah Morell, and Lars Rohwedder. The submodular santa claus problem. *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 616–640, 2025.

4 Nikhil Bansal and Maxim Sviridenko. The santa claus problem. *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 31–40, 2006.

5 Siddharth Barman and Sanath Kumar Krishnamurthy. Approximation algorithms for maximin fair division. *ACM Transactions on Economics and Computation (TEAC)*, 8(1):1–28, 2020.

6 Ivona Bezáková and Varsha Dani. Allocating indivisible goods. *ACM SIGecom Exchanges*, 5(3):11–18, 2005.

7 Deeparnab Chakrabarty, Julia Chuzhoy, and Sanjeev Khanna. On allocating goods to maximize fairness. *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 107–116, 2009.

8 Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 575–584, 2010.

9 Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 783–792, 2011.

10 Siu-Wing Cheng and Yuchen Mao. Restricted max-min allocation: Approximation and integrality gap. *arXiv preprint arXiv:1905.06084*, 2019.

11 Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

12 Mohammad Ghodsi, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods: Improvements and generalizations. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 539–556, 2018.

13 Michel X Goemans, Nicholas JA Harvey, Satoru Iwata, and Vahab Mirrokni. Approximating submodular functions everywhere. *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 535–544, 2009.

14 Daniel Golovin. *Max-min fair allocation of indivisible goods*. School of Computer Science, Carnegie Mellon University, (2005).

15 Penny Haxell and Tibor Szabó. Improved integrality gap in max–min allocation, or, topology at the north pole. *Combinatorica*, 45(2):1–38, 2025.

16 Dorit S Hochbaum and David B Shmoys. Using dual approximation algorithms for scheduling problems theoretical and practical results. *Journal of the ACM (JACM)*, 34(1):144–162, 1987.

17 Subhash Khot, Richard J Lipton, Evangelos Markakis, and Aranyak Mehta. Inapproximability results for combinatorial auctions with submodular utility functions. *Algorithmica*, 52:3–18, 2008.

18 Andreas Krause, Ram Rajagopal, Anupam Gupta, and Carlos Guestrin. Simultaneous placement and scheduling of sensors. *2009 International Conference on Information Processing in Sensor Networks*, pages 181–192, 2009.

19 David Kurokawa, Ariel D Procaccia, and Nisarg Shah. Leximin allocations in the real world. *ACM Transactions on Economics and Computation (TEAC)*, 6(3-4):1–24, 2018.

20 Frank M Liang. A lower bound for on-line bin packing. *Information processing letters*, 10(2):76–79, 1980.

21 Vahab Mirrokni, Michael Schapira, and Jan Vondrák. Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. *Proceedings of the 9th ACM conference on Electronic commerce*, pages 70–77, 2008.

22 Lukáš Poláček and Ola Svensson. Quasi-polynomial local search for restricted max-min fair allocation. *ACM Transactions on Algorithms (TALG)*, 12(2):1–13, 2015.

23 Sartaj Sahni. Approximate algorithms for the 0/1 knapsack problem. *Journal of the ACM (JACM)*, 22(1):115–124, 1975.

24 Steven S Seiden and Gerhard J Woeginger. The two-dimensional cutting stock problem revisited. *Mathematical Programming*, 102(3):519–530, 2005.

25 Ola Svensson. Santa claus schedules jobs on unrelated machines. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 617–626, 2011.

26 Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.

27 Zoya Svitkina and Lisa Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM Journal on Computing*, 40(6):1715–1737, 2011.

28 Gilad Ben Uziahu and Uriel Feige. On fair allocation of indivisible goods to submodular agents. *arXiv preprint arXiv:2303.12444*, 2023.

29 Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 67–74, 2008.

30 Gerhard J Woeginger. A polynomial-time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters*, 20(4):149–154, 1997.

Appendix A Basic Facts about Submodular Functions

Let $f : 2^J \to \mathbb{R}$ be submodular and monotone.

▶ **Lemma A.1.** *Consider a new item $z \notin J$. For any $t \in \mathbb{R}$, define the function $f_t : 2^{J \cup \{z\}} \to \mathbb{R}$ 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 is clear. For submodularity,

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

◀

The following statements are folklore.

▶ **Observation A.2.** *The restriction of f to any subset $J' \subseteq J$ is submodular and monotone.*

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

▶ **Observation A.5.** $f(S \cup T) = f(S) + \Delta_f(T \mid S)$.

▶ **Observation A.6.** *For $S \subseteq T$, $\Delta_f(S \mid T) = 0$.*

Appendix B Truncated Max-Sum Greedy

In the following, we describe an instance where $\mathsf{OPT} = 5$ but the greedy algorithm achieves only $2 + \delta$ for a small δ, independently of the choice of α. We first show that it is sufficient to find a certain bad example in an instance with an additive valuation function.

▶ **Lemma B.1.** *Assume there is an instance $\mathcal{I}_{\mathsf{additive}}$ with an additive valuation function and a $\delta \geq 0$ with the following properties:*

- *There exists a partial allocation A^* such that some item j with $f(j) > 0$ is* not *allocated in A^* and $f_{\mathsf{min}}(A^*) \geq 3 - \delta$*
- *If $\alpha \cdot \mathsf{OPT} > 2 + \delta$, greedy can produce an allocation A where all items are allocated.*

Then, there exists an instance $\mathcal{I}_{\mathsf{submodular}}$ with a submodular valuation function where the optimal solution achieves at least 5 and independently of the choice of α, greedy can produce an allocation A with $f_{\mathsf{min}}(A) \leq \frac{2+\delta}{5}$.

Proof. To create $\mathcal{I}_{\mathsf{submodular}}$, start by copying $\mathcal{I}_{\mathsf{additive}}$ $\lceil \frac{5}{f(j)} \rceil$ many times (i.e. create new items and new players, while maintaining the additive weights of all items). Let P' be the players and J' be the items of the resulting instance. A valid run of greedy simulates each step of the run on $\mathcal{I}_{\mathsf{additive}}$ on every copy in "parallel". Therefore, the resulting allocation A' still allocates all items. Clearly, there still exists an allocation A'^* with $f_{\mathsf{min}}(A'^*) \geq 5$.

Let $s = \min_{j' \in J'} f(j')$ (we can assume $s > 0$, otherwise we could have discarded the item j' with $f(j') = 0$), and let $0 < s' \leq s$ divide $2 + \delta$. Create a new item z^*, and for $p \in P'$ and $i \in [N := \frac{2+\delta}{s'}]$, create an item z_i^p.

Let us define a new valuation function f' on all items. Set $\Delta_{f'}(z_i^p \mid S) = s'$ if $z_i^{p'}, z^* \notin S$ for all p', and set $\Delta_{f'}(z_i^p \mid S) = 0$ otherwise. Set $\Delta_{f'}(z^* \mid S) = 2 + \delta - |\{j \mid \exists p : z_i^p \in S\}|$. For all $j' \in J'$, set $\Delta_{f'}(j' \mid S) = f(j')$ for all sets S. It is straightforward to check that f' is a coverage function and thus submodular.

Our instance $\mathcal{I}_{\mathsf{submodular}}$ consists of J' together with z^* and all z_i^p, valuation function f' and players $P' \cup \{p^*\}$ for a single new player p^*. First, let us show that there exists an allocation B^* with $f_{\mathsf{min}}(B^*) \geq 5$. For each player $p \in P'$, set $B_p^* = A_p'^* \cup \{z_j^p \mid j \in [N]\}$. Then, $f(B_p^*) = f(A_p'^*) + N \cdot s' \geq 3 - \delta + 2 + \delta = 5$. To the special player p^*, we give all copies of item j. Since there are $\lceil \frac{5}{f(j)} \rceil$ many copies, we get $f(B_p^*) \geq 5$.

Now, let us argue that greedy can produce an allocation B with $f_{\mathsf{min}}(B) \leq 2 + \delta$. We can assume $\alpha \cdot \mathsf{OPT} \geq 2 + \delta + \varepsilon$ for some $\varepsilon > 0$, since otherwise it is easy to construct an instance where greedy achieves at most $2 + \delta$ (e.g., many items of additive value $2 + \delta$). First, notice that greedy can start by allocating all additive items as in A', since the z_i^p-items have smaller marginal contributions. At this point, p^* has not received any item. Since for the remaining items $f(z^* \cup \bigcup_{p,j} z_i^p) = 2 + \delta$, in the final allocation B, player p^* can achieve at most value $2 + \delta$. This finishes the proof.

◀

Now we are left with exhibiting an instance that satisfies the conditions of the lemma for a δ as small as possible. Fix some $N \in \mathbb{N}^+$ and let $s_1, \ldots, s_N$ be the first N members of the Sylvester sequence $s_n = 1 + \prod_{i=1}^{n-1} s_i$. Let $m = 2 \cdot (s_N - 1)$ be the number of players and J contain the following items:

- $m/2$ items of size $2 + \delta$
- m items of size $\frac{3-\delta}{2}$
- $m/2$ items of size $\frac{1+3\delta}{2(s_i-1)}$ for each $i \in [N-1]$
- $m/2 + 1$ items of size $\frac{1+3\delta}{2(s_N-1)}$.

The following is a valid greedy allocation for threshold value $\alpha \cdot \mathsf{OPT} > 2 + \delta$.

- $m/2$ players receive $2 + \delta$ and $\frac{3-\delta}{2}$
- For each $i \in [N-1]$, $\frac{m}{2s_i}$ players receive $\frac{3-\delta}{2}$ and s_i items of size $\frac{1+3\delta}{2(s_i-1)}$
- one player receives $\frac{3-\delta}{2}$ and s_N items of size $\frac{1+3\delta}{2(s_N-1)}$.

The number of items of size $\frac{3-\delta}{2}$ that are used is

$$\frac{m}{2} + \Big(\sum_{i=1}^{N-1} \frac{m}{2s_i}\Big) + 1 = \frac{m}{2} + \frac{m}{2} \cdot \Big(\sum_{i=1}^{N-1} \frac{1}{s_i}\Big) + 1 = \frac{m}{2} + \frac{m}{2} \cdot \Big(1 - \frac{1}{s_N - 1}\Big) + 1 = m$$

where the equality $\sum_{i=1}^{N-1} \frac{1}{s_i} = 1 - \frac{1}{s_N - 1}$ is known and easy to show. One can check that each player gets one of the m largest items, that all items got allocated and that if we remove one item from a player, the sum of the remaining item sizes is at most $2 + \delta < \alpha \cdot \mathsf{OPT}$. The following is another solution:

- $m/2$ players receive 2 items of size $\frac{3-\delta}{2}$
- $m/2$ players receive $2 + \delta$ and one item of size $\frac{1+3\delta}{2(s_i-1)}$ for each $i \in [N]$.

Notice that one item of size $\frac{1+3\delta}{2(s_N-1)}$ is not allocated. The first half of the players achieves value $3 - \delta$. For the second half to also achieve the same value, we need

$$2 + \delta + \frac{1 + 3\delta}{2} \cdot S = 3 - \delta$$

where $S = \sum_{i=1}^{N} \frac{1}{s_i - 1}$. Therefore we can apply Lemma B.1 with

$$\delta = \frac{2 - S}{4 + 3S}$$

When N goes to infinity, $S \approx 1.691$. Thus the ratio between the greedy solution and the optimal solution is at least

$$\frac{2+\delta}{5} = \frac{2+S}{4+3S} \approx 0.4068.$$

We believe that this is the worst possible ratio achievable by the greedy algorithm.

Appendix C Lower bound on the integrality gap

▶ **Proposition 1.4.** *The integrality gap of the configuration LP with identical submodular valuations is at least* $4/3$*, even with* 6 *items and* 3 *players.*

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. ◀