

# Computing Approximately Proportional Allocations of Indivisible Goods: Beyond Additive and Monotone Valuations

Martin Jupakkal Andersen, Ioannis Caragiannis, Anders Bo Ipsen, and Alexander Søltoft

Department of Computer Science, Aarhus University, Åbogade 34, 8200 Aarhus N, Denmark

## Abstract

Although approximate notions of envy-freeness—such as envy-freeness up to one good (EF1)—have been extensively studied for indivisible goods, the seemingly simpler fairness concept of proportionality up to one good (PROP1) has received far less attention. For additive valuations, every EF1 allocation is PROP1, and well-known algorithms such as Round-Robin and Envy-Cycle Elimination compute such allocations in polynomial time. PROP1 is also compatible with Pareto efficiency, as maximum Nash welfare allocations are EF1 and hence PROP1.

We ask whether these favorable properties extend to non-additive valuations. We study a broad class of allocation instances with *satiating goods*, where agents have non-negative valuation functions that need not be monotone, allowing for negative marginal values. We present the following results:

- EF1 implies PROP1 for submodular valuations over satiating goods, ensuring existence and efficient computation via Envy-Cycle Elimination for monotone submodular valuations;
- Round-robin computes a partial PROP1 allocation after the second-to-last round for satiating submodular goods and a complete PROP1 for submodular monotone valuations;
- PROP1 allocations for satiating subadditive goods can be computed in polynomial-time;
- Maximum Nash welfare allocations are PROP1 for monotone submodular goods, revealing yet another facet of their “unreasonable fairness.”

## 1 Introduction

Proportionality (Steinhaus 1948) is the most important share-based fairness concept in the fair division literature. An allocation is proportional if each of the  $n$  agents gets at least a  $1/n$ -th of all items in terms of value. Unfortunately, for allocation problems with indivisible items, proportional allocations may not exist. The relaxed concept of proportionality up to some item (PROP1), introduced by Conitzer, Freeman, and Shah (2017), comes to address this limitation. PROP1 is well understood in allocation instances with goods and agents with additive valuations. However, our understanding of it in settings with more general agent valuations is very limited. We aim to fill this gap in this work.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

PROP1 relaxes proportionality in a similar spirit to how EF1 (envy-freeness up to some item; introduced by Budish (2011)) relaxes envy-freeness. As proved by Conitzer, Freeman, and Shah (2017), for additive valuations, EF1 implies PROP1; thus, several results that guarantee the existence and efficient computation of EF1 allocations trivially imply the existence and efficient computation of PROP1 allocations as well. For example, the well-known Envy-Cycle Elimination algorithm of Lipton et al. (2004) and the folklore Round-Robin algorithm produce EF1 allocations while the maximum Nash welfare allocations are EF1 and Pareto-optimal (Caragiannis et al. 2019). As a corollary of the EF1-to-PROP1 implication for additive valuations, we can replace EF1 with PROP1 in these three results.

Furthermore, the celebrated Envy-Cycle Elimination algorithm works on allocation instances with monotone goods and computes an EF1 allocation. So, one would even hope to use it to get PROP1 allocations on instances with monotone valuations. Unfortunately, such a positive result is not possible. For example, consider the instance with two agents and three items such that both agents have valuation 2 for the whole set of items and value 0 for every strict subset of it. The valuation function is clearly monotone. However, in any allocation, some agent will get only one item and her value will be 0 even after adding one additional item to her bundle, i.e., below the proportionality threshold.

The above example implies that for sufficiently general (e.g., *super-additive*) valuations, EF1 allocations are not PROP1. What is the broader class of allocation instances in which EF1 implies PROP1? Can we then use well-known algorithms for EF1 (such as the Envy-Cycle Elimination and Round-Robin) to produce PROP1 allocations? What is the broader class of allocation instances in which a PROP1 allocation can be computed in polynomial time? What is the broader class of allocation instances in which PROP1 is compatible with Pareto-optimality? These are the questions we study in this paper, making substantial progress towards understanding PROP1.

**Our contribution.** We address the questions above and present a list of new results on PROP1 allocations for instances with more general than additive valuation functions. At the conceptual level, we consider valuation functions over *satiating goods*. Such functions return non-negative

values for bundles but the marginal value of adding a good to a bundle can be negative. An agent typically has positive value for getting a single item but adding this item into a bundle that already contains some other items can decrease her value. The definitions of submodular and subadditive valuation functions are naturally extended to satiating goods. Submodularity means non-increasing marginal value while subadditivity means that the valuation for a bundle of items is not higher than the sum of valuations for any two disjoint subsets of it.

Our first technical result is that EF1 allocations are also PROP1 in allocation instances with satiating submodular goods. The implication is also true for allocation instances with two agents and satiating subadditive valuations but is not true for allocation instances with more agents and slightly more general than monotone submodular valuations. As a corollary, we get that the Envy-Cycle Elimination algorithm computes PROP1 allocations for instances with monotone submodular goods. These results appear in Section 3.

Next, in Section 4, we study the Round-Robin algorithm. Round-Robin is well-known to produce EF1 allocation for additive goods but fails to do so for more general valuation functions; see Amanatidis et al. (2023b). Hence, the implication between EF1 and PROP1 does not have any implication for Round-Robin in non-additive instances. Somewhat surprisingly, we prove that Round-Robin does produce a PROP1 allocation on instances with monotone submodular goods. Actually, it almost does so for satiating submodular goods as well. The partial allocation it computes after its second-to-last round is PROP1. These results are best possible. The final allocation of Round-Robin may not be PROP1 when applied on allocation instances with satiating submodular goods or with monotone valuations that are slightly more general than submodular.

Our strongest algorithmic result is a new algorithm that computes a PROP1 allocation for satiating subadditive goods. The algorithm starts with an arbitrary allocation and gradually satisfies the PROP1 conditions for more and more agents by repeatedly moving items from bundle to bundle. The proof of correctness uses a nice potential function argument. The algorithm can be combined with Round-Robin to give a considerably faster algorithm for satiating submodular goods. These results are presented in Section 5.

Finally, we address the question of whether PROP1 is compatible with Pareto-optimality. Our main positive result is another facet of the “unreasonable fairness” of maximum Nash welfare allocations. Such allocations are PROP1 and Pareto-optimal on instances with monotone submodular goods. This is the broadest class of monotone instances in which PROP1 and Pareto-optimality are always compatible. We present instances with slightly more general monotone valuation functions than submodular where no PROP1 allocation is Pareto-optimal. These results appear in Section 6.

We continue by discussing the related literature in the rest of this section. We present useful preliminary definitions in Section 2 that are necessary for the presentation of our technical results in Sections 3-6, and conclude with open problems in Section 7.

**Related work.** PROP1 was introduced by Conitzer, Freeman, and Shah (2017). They considered additive valuation functions and observed, among other results, that EF1 allocations are PROP1. Three papers by Aziz, Moulin, and Sandomirskiy (2020), Barman and Krishnamurthy (2019), and McGlaughlin and Garg (2020) consider the compatibility of PROP1 and Pareto-optimality and present related algorithmic results. For instances with additive goods, PROP1 seems to be the simplest fairness property; the recent work by Garg and Sharma (2025) summarize how PROP1 is implied by other fairness properties (see also the related software at <https://sharmaeklavaya2.github.io/cpigjs/fairDiv/>).

Non-additive valuation functions have been considered extensively recently. Submodular valuation functions are central in the study of allocation problems since the seminal work of Lehmann, Lehmann, and Nisan (2006), who also introduced the class XOS (or fractionally subadditive valuations). In fair division, Amanatidis et al. (2023b) and Barman and Krishnamurthy (2020) have studied the Round-Robin protocol on allocation instances with submodular valuations. Subadditive valuations have been considered in relation to approximate versions of the fairness notions EFX (Chaudhury, Garg, and Mehta 2021; Feldman, Murras, and Ponitka 2024; Barman and Suzuki 2024) and MMS (Seddighin and Seddighin 2025); see also the survey by Amanatidis et al. (2023a). Subadditive valuations have been important in the study of utilitarian and Nash welfare maximization by, e.g., Feige (2009) and Dobzinski et al. (2024), respectively.

The above results refer to goods, i.e., monotone valuation functions with non-negative marginals. In fair division, non-monotonicity has been studied in the model of combined goods and chores of Aziz et al. (2022). These valuation functions are additive but an item can have positive value for an agent and negative to another. Our definition of satiating valuations has non-negative valuations but allows for negative marginals. Similar valuations (with possibly negative values for non-empty bundles that do not contain all items) have been considered very recently by Barman and Verma (2025), in relation to the equitability fairness concept. The literature on optimization of set functions has focused extensively on satiating submodular valuations, starting with the work of Feige, Mirkroki, and Vondrák (2011); see also the related survey by Krause and Golovin (2014).

## 2 Preliminaries

We consider *allocation instances* in which a set  $M$  of  $m$  goods (or items) has to be allocated to  $n$  agents. We identify the agents using the positive integers in  $[n] = \{1, 2, \dots, n\}$ . Each agent has a *valuation function* which returns the value the agent has for each set (or *bundle*) of goods. We denote by  $v_i$  the valuation function of agent  $i \in [n]$ ; it is normalized with  $v_i(\emptyset) = 0$  and takes non-negative values, i.e.,  $v_i(S) \geq 0$ , for every bundle  $S \subseteq M$  of goods. When  $S$  is a singleton with, say,  $S = \{o\}$ , we simplify notation and use  $v_i(o)$  instead of  $v_i(\{o\})$ . The valuation function  $v_i$  is called *additive* if  $v_i(S) = \sum_{o \in S} v_i(o)$  for every set of goods  $S \subseteq M$ .

For two disjoint bundles of goods  $S$  and  $T$ , we use the notation  $v_i(T|S) := v_i(T \cup S) - v_i(S)$  to denote the *marginal value* the bundle  $T$  has for agent  $i \in [n]$  when it is added to the bundle  $S$ . When  $T$  is a singleton with, say,  $T = \{o\}$ , we simplify notation to  $v_i(o|S)$  (instead of  $v_i(\{o\}|S)$ ).

The valuation function  $v_i$  is called *submodular* if  $v_i(S \cup T) + v_i(S \cap T) \leq v_i(S) + v_i(T)$  for every sets  $S, T \subseteq M$ . A submodular valuation function satisfies  $v_i(o|S) \geq v_i(o|T)$  for every two sets of goods  $S$  and  $T$  with  $S \subseteq T \subseteq M$  and good  $o \notin T$ .

Our definition for valuation functions allows for marginal values to have any sign. We use the term *monotone submodular goods* for submodular valuation functions with non-negative marginal values and *satiating submodular goods* for general submodular valuation functions.

The next claim applies to the most general definition of satiating submodular goods and gives an equivalent definition of submodularity, which we use in our proofs.

**Claim 1.** *The submodular valuation function  $v_i$  satisfies  $v_i(X|S) \geq v_i(X|T)$  for every set of goods  $X \subseteq M$  and every two sets of goods  $S$  and  $T$  with  $S \subseteq T \subseteq M$  and  $X \cap T = \emptyset$ .*

Very often in our proofs, we use the following property.

**Claim 2.** *The submodular valuation function  $v_i$  satisfies  $v_i(X \cup T|S) \leq v_i(X|S) + v_i(T|S)$  for every mutually disjoint sets of goods  $X, S$ , and  $T$ . Hence, it also satisfies  $v_i(T|S) \leq \sum_{o \in T} v_i(o|S)$  for every two disjoint sets of goods  $S$  and  $T$ .*

*Proof.* To prove the first part of Claim 2, we use the definition of marginal value and submodularity (Claim 1) to get  $v_i(X \cup T|S) = v_i(X|S) + v_i(T|X \cup S) \leq v_i(X|S) + v_i(T|S)$ .  $\square$

The valuation function  $v_i$  is called *subadditive* if  $v_i(S \cup T) \leq v_i(S) + v_i(T)$  for every two sets of goods  $S, T \subseteq M$ . Again, we use the terms *monotone* and *satiating subadditive goods* to distinguish between subadditive valuation functions with non-negative marginal and general subadditive valuation functions, respectively.

Every submodular valuation function is also subadditive. In some proofs, we use (monotone) *XOS* valuation functions defined as follows. A valuation function  $v_i(\cdot)$  over bundles of items is *XOS* if there are additive valuation functions  $f_1(\cdot), f_2(\cdot), \dots, f_k(\cdot)$  such that  $v_i(S) = \max_{j \in [k]} f_j(S)$ . A monotone submodular valuation function is also *XOS* and a *XOS* valuation function is also monotone subadditive. Figure 1 summarizes the different valuation functions used in the paper.

An allocation  $A = (A_1, A_2, \dots, A_n)$  is a disjoint partition of the set of goods  $M$  to the  $n$  agents. For  $i \in [n]$ , the set  $A_i$  indicates the bundle of goods allocated to agent  $i$ . The allocation  $A$  is *envy-free* if  $v_i(A_i) \geq v_i(A_j)$  for every pair of agents  $i, j \in [n]$  (i.e., agent  $i$  weakly prefers her own bundle to the bundle of agent  $j$ ) and is *proportional* if  $v_i(A_i) \geq \frac{1}{n} \cdot v_i(M)$  (i.e., agent  $i$  has a value for her bundle that is at least as high as her *proportionality threshold*). An allocation is *envy-free up to some good* (EF1) if for every two agents  $i, j \in [n]$ , it is either  $v_i(A_i) \geq v_i(A_j)$

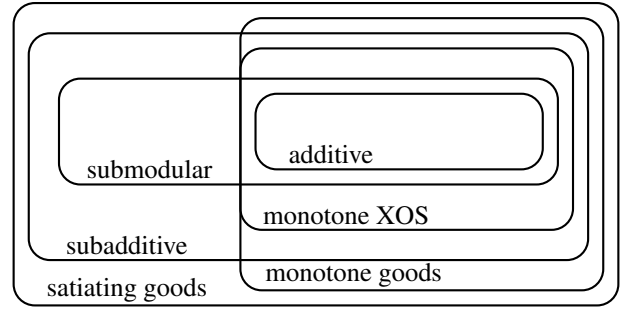


Figure 1: Relation between the valuation functions over goods used in the paper.

or there exists a good  $g$  in the bundle of agent  $j$  such that  $v_i(A_i) \geq v_i(A_j \setminus \{g\})$ . An allocation is *proportional up to some good* (PROP1) if for every agent  $i \in [n]$ , it is either  $v_i(A_i) \geq \frac{1}{n} \cdot v_i(M)$ , or there exists a good  $g$  not allocated to agent  $i$  such that  $v_i(A_i \cup \{g\}) \geq \frac{1}{n} \cdot v_i(M)$ . These definitions apply to the most general case of satiating goods. For monotone goods, we trivially have  $v_i(A_i \cup \{g\}) \geq v_i(A_i)$  and the first condition in the definition of EF1 and PROP1 is redundant.

### 3 Does EF1 Imply PROP1?

Our first technical result generalizes the well-known fact (see Conitzer, Freeman, and Shah 2017, Lemma 3.6) that EF1 allocations are also PROP1 to allocation instances with non-additive and non-monotone valuations.

**Theorem 3.** *In any allocation instance with satiating submodular goods, an EF1 allocation is also PROP1.*

*Proof.* Consider an allocation instance with  $n$  agents having submodular valuations for subsets of a set of satiating goods  $M$ , and let  $A = (A_1, A_2, \dots, A_n)$  be an EF1 allocation.

Let  $i \in [n]$  be an agent. Define  $\tilde{N} := \{j \in [n] : v_i(A_i) \geq v_i(A_j)\}$  and observe that  $\tilde{N}$  is non-empty since  $i \in \tilde{N}$ . If  $\tilde{N} = [n]$ , then applying the condition  $v_i(A_i) \geq v_i(A_j)$  for  $j \in [n]$ , we get

$$n \cdot v_i(A_i) \geq \sum_{j \in [n]} v_i(A_j) \geq v_i(M), \quad (1)$$

completing the proof. The second inequality in Equation (1) follows by the subadditivity of the valuation function  $v_i$ .

Now, assume that  $\tilde{N} \neq [n]$ . For each agent  $j \in [n] \setminus \tilde{N}$ , let  $o_j$  be an (arbitrary) item in the bundle  $A_j$  such that  $v_i(A_i) \geq v_i(A_j \setminus \{o_j\})$ ; since allocation  $A$  is EF1 and  $v_i(A_i) < v_i(A_j)$ , we get  $v_i(A_j) > 0$ , meaning that such a good clearly exists. Let  $O := \{o_j : j \in [n] \setminus \tilde{N}\}$  and set  $o^* := \arg \max_{o \in O} v_i(o|A_i)$ . We will prove the theorem by showing that  $v_i(M) \leq n \cdot \max\{v_i(A_i), v_i(A_i \cup \{o^*\})\}$ .

Notice that the set  $M \setminus O$  is the disjoint union of the bundles  $A_j$  for  $j \in \tilde{N}$  and  $A_j \setminus \{o_j\}$  for  $j \in [n] \setminus \tilde{N}$ . By the subadditivity of the valuation function  $v_i$ , we have

$$v_i(M \setminus O) \leq \sum_{j \in \tilde{N}} v_i(A_j) + \sum_{j \in [n] \setminus \tilde{N}} v_i(A_j \setminus \{o_j\})$$

$$\leq n \cdot v_i(A_i). \quad (2)$$

The last inequality follows since the allocation  $A$  is EF1.

By the definition of set  $O$ , we have  $A_i \cap O = \emptyset$ . Thus,  $A_i \subseteq M \setminus O$  and

$$\begin{aligned} v_i(O|M \setminus O) &\leq v_i(O|A_i) \leq \sum_{o \in O} v_i(o|A_i) \\ &\leq \sum_{o \in O} v_i(o^*|A_i) \\ &= (n - |\tilde{N}|) \cdot v_i(o^*|A_i). \end{aligned} \quad (3)$$

The first inequality follows by the submodularity of valuation function  $v_i$  (Claim 1), the second one by Claim 2, and the third one by the definition of  $o^*$ .

By Equations (2) and (3), we have

$$\begin{aligned} v_i(M) &= v_i(M \setminus O) + v_i(O|M \setminus O) \\ &\leq n \cdot v_i(A_i) + (n - |\tilde{N}|) \cdot v_i(o^*|A_i). \end{aligned} \quad (4)$$

Now, if  $v_i(o^*|A_i) < 0$ , Equation (4) immediately implies  $v_i(M) \leq n \cdot v_i(A_i)$ . Otherwise, we get

$$\begin{aligned} v_i(M) &\leq n \cdot v_i(A_i) + (n - |\tilde{N}|) \cdot v_i(o^*|A_i) \\ &\leq n \cdot v_i(A_i) + n \cdot v_i(o^*|A_i) \\ &= n \cdot v_i(A_i \cup \{o^*\}), \end{aligned}$$

again completing the proof.  $\square$

Theorem 3 has important algorithmic implications. Using the well-known result of Lipton et al. (2004) that the Envy-Cycle Elimination algorithm produces EF1 allocations when applied to monotone allocation instances, we get the following corollary.

**Corollary 4.** *Given an allocation instance with monotone submodular goods, the Envy-Cycle Elimination algorithm returns a PROP1 allocation.*

We will extend and significantly improve Corollary 4 in Sections 4 and 5, respectively.

The result in Theorem 3 is best possible. The proof of the next theorem uses an allocation instance with monotone valuations that are slightly more general than submodular.

**Theorem 5.** *There exists an allocation instance with monotone XOS goods that has an EF1 allocation that is not PROP1.*

*Proof.* Consider an allocation instance with three agents and seven items. Each agent has the valuation function  $v$  depicted in Table 1. Notice that  $v(S)$  depends only on the cardinality of  $S$ . We can verify that  $v$  is XOS. Indeed, using the positive integers in [7] to identify the items, we have  $v(S) = \max_{j=0,1,\dots,7} \sum_{g \in S} f_j(g)$  with  $f_0(g) = 1$  for  $g \in [7]$ ,  $f_g(g) = 2$  for  $g \in [7]$ , and  $f_t(g) = 0$  for  $t, g \in [7]$  with  $t \neq g$ .

Now, consider an allocation  $A = (A_1, A_2, A_3)$  with  $|A_1| = 1$ ,  $|A_2| = |A_3| = 3$ . Clearly, agents 2 and 3 are non-envious. For every strict subset of bundles  $A_2$  and  $A_3$ , agent 1 has value 2, i.e., equal to her value  $v_1(A_1)$  in allocation  $A$ . Hence, allocation  $A$  is EF1. It is not PROP1, though. Agent

$ S $	1	2	3	4	5	6	7
$v(S)$	2	2	3	4	5	6	7

Table 1: The valuation function of the agents in the instance used in the proof of Theorem 5.

1 has still value 2 for any bundle obtained by adding an extra item to her bundle  $A_1$ , while her proportionality threshold is  $7/3$ .  $\square$

We remark that the proof of Theorem 5 uses an instance with three agents. Interestingly, for two-agent allocation instances and satiating subadditive goods, EF1 implies PROP1 as the next statement indicates.

**Theorem 6.** *In allocation instances with satiating subadditive goods and two agents, any EF1 allocation is also PROP1.*

*Proof.* Consider an allocation instance with two agents having subadditive valuations for subsets of a set of satiating goods and let  $A = (A_1, A_2)$  be an EF1 allocation. Without loss of generality, we will focus on agent 1 and prove that allocation  $A$  satisfies the PROP1 condition for this agent.

First, assume that  $v_1(A_1) \geq v_1(A_2)$ . Using subadditivity and this inequality, we get

$$v_1(A_1 \cup A_2) \leq v_1(A_1) + v_1(A_2) \leq 2 \cdot v_1(A_1),$$

completing the proof. Otherwise, due to the fact that allocation  $A$  is EF1, there exists an item  $g$  in the bundle  $A_2$ , such that  $v_1(A_1) \geq v_1(A_2 \setminus \{g\})$ . Using subadditivity and this inequality, we get

$$\begin{aligned} v_1(A_1 \cup A_2) &\leq v_1(A_2 \setminus \{g\}) + v_1(A_1 \cup \{g\}) \\ &\leq v_1(A_1) + v_1(A_1 \cup \{g\}) \\ &\leq 2 \cdot \max\{v_1(A_1), v_1(A_1 \cup \{g\})\}, \end{aligned}$$

again completing the proof.  $\square$

## 4 Does Round-Robin Compute PROP1 Allocations?

We now focus on the Round-Robin protocol, arguably the simplest algorithm for allocating indivisible goods. Round-Robin starts with an empty allocation and runs in rounds. In each round, the agents act in a fixed order. When it is the turn of an agent to act, they pick the best item that is still available, i.e., the available item that has the maximum marginal value for the agent. For satiating goods, this marginal value can be negative.

Our main result in this section (Corollary 9) applies to allocation instances with monotone submodular goods. For this case, Amanatidis et al. (2023b) have proven that the allocation returned by Round-Robin is not always EF1. Somewhat surprisingly, we prove that it is PROP1, using different arguments than those developed in Section 3.

For satiating submodular goods, Round-Robin achieves PROP1 only partially, as the next statement indicates.

**Theorem 7.** *In every allocation instance with satiating submodular goods, the partial allocation produced by Round-Robin after the second-to-last round is PROP1.*

One might think that the fact that Theorem 7 refers to the partial allocation computed at the end of the second-to-last round, as opposed to the final complete allocation returned by Round-Robin, is just a weakness in our analysis. This is not the case, though, as the next statement shows.

**Theorem 8.** *There exists an allocation instance with satiating submodular goods for which Round-Robin does not produce a PROP1 allocation.*

*Proof.* We use an allocation instance with two agents and six items  $a, b, c, d, g_1$ , and  $g_2$ . Agent 1 has an additive valuation function, with value 1 for items  $a, b$ , and  $c$  and value 0 for items  $d, g_1$ , and  $g_2$ . Agent 2 has the valuation function defined in Table 2. Items  $g_1, g_2, a, b$ , and  $c$  have constant marginal values  $v(\cdot|S)$  of 3, 3, 2, 2, and 2, respectively, when added to a bundle  $S$  not containing item  $d$ . Item  $d$  has marginal value  $v_2(d|S)$  equal to  $-5$  if set  $S$  contains both  $g_1$  and  $g_2$ , and equal to 2 otherwise. As marginal values are non-increasing but possibly negative, this is an instance of satiating submodular goods.

item $o$	bundle $S$	$v_2(o S)$
$g_1, g_2$	$d \notin S$	3
$a, b, c$	$d \notin S$	2
$d$	$\{g_1, g_2\} \not\subseteq S$	2
$d$	$\{g_1, g_2\} \subseteq S$	$-5$

Table 2: The valuation function of agent 2 in the proof of Theorem 8.

By applying Round-Robin, agent 1 picks the items  $a, b$ , and  $c$  in the first three rounds. Agent 2 picks items  $g_1$  and  $g_2$  in the first two round and is left with item  $d$  in the third round. We have  $v_2(M) = 7$  while  $v_2(\{g_1, g_2, d, o\}) = 3 < v_2(M)/2$  for every  $o \in \{a, b, c\}$ , violating the PROP1 condition.  $\square$

Our next positive result for monotone submodular goods follows easily using Theorem 7 and by arguing about the additional value the agents get in the last round of Round-Robin.

**Corollary 9.** *For every allocation instance with monotone submodular goods, Round-Robin returns a PROP1 allocation.*

*Proof.* Consider an allocation instance with  $n$  agents having submodular valuations for subsets of a set of satiating goods  $M$ . Let  $L = \lceil |M|/n \rceil$  be the number of rounds executed by Round-Robin. For agent  $i \in [n]$  and  $k \in [L]$ , denote by  $A_i^k$  the set of items allocated by Round-Robin to agent  $i$  in rounds  $1, 2, \dots, k$  and let  $A = (A_1, A_2, \dots, A_n)$  be the final complete allocation. By Theorem 7, we get that either  $v_i(M) \leq n \cdot v_i(A_i^{L-1})$  or there exists an item  $g \notin A_i^{L-1}$  such that  $v_i(M) \leq n \cdot v_i(A_i^{L-1} \cup \{g\})$ .

If  $v_i(M) \leq n \cdot v_i(A_i^{L-1})$ , monotonicity yields  $v_i(A_i^{L-1}) \leq v_i(A_i)$  and, hence,  $v_i(M) \leq n \cdot v_i(A_i)$  as well. If  $v_i(M) \leq n \cdot v_i(A_i^{L-1} \cup \{g\})$ , we distinguish between two cases. First, if item  $g$  is allocated to agent  $i$  in the last

round by Round-Robin, we have  $v_i(A_i) = v_i(A_i^{L-1} \cup \{g\})$  and, hence,  $v_i(M) \leq n \cdot v_i(A_i)$ . Otherwise, monotonicity implies that  $v_i(A_i^{L-1} \cup \{g\}) \leq v_i(A_i \cup \{g\})$  and, hence,  $v_i(M) \leq n \cdot v_i(A_i \cup \{g\})$ , completing the proof.  $\square$

The result in Corollary 9 is best possible. The proof of the next theorem uses an allocation instance with monotone valuations that are slightly more general than submodular.

**Theorem 10.** *There exists an allocation instance with monotone XOS goods for which Round-Robin does not produce a PROP1 allocation.*

*Proof.* We prove the theorem using an allocation instance with two agents and six items  $a, b, c, d, e$ , and  $f$ . Agent 1 has the additive valuation function depicted in Table 3.

item	$a$	$b$	$c$	$d$	$e$	$f$
$v_1(\cdot)$	3	2	1	0	0	0

Table 3: The additive valuation function of agent 1 used in the proof of Theorem 10.

Agent 2 has an XOS valuation function  $v_2$  that uses the two additive valuation functions  $f_1$  and  $f_2$  depicted in Table 4. In particular,  $v_2(S) = \max\{f_1(S), f_2(S)\}$  for every  $S \subseteq \{a, b, c, d, e, f\}$ .

item	$a$	$b$	$c$	$d$	$e$	$f$
$f_1(\cdot)$	5	5	5	0	0	4
$f_2(\cdot)$	0	0	0	6	2	1

Table 4: The additive valuation functions that are used in the definition of the XOS valuation function  $v_2$  of agent 2 in the proof of Theorem 10.

Agent 2 has item  $d$  as the most valuable singleton. Then, the marginal valuation  $v_2(o|d)$  is maximized for item  $e$  with  $v_2(e|d) = 2$ . Also, the marginal valuation  $v_2(o|\{d, e\})$  is maximized to 1 for item  $f$ ; any other marginal is equal to 0. So, in any execution of Round-Robin, agent 2 will receive the items  $d, e$ , and  $f$  (in this order) if they are available. Clearly, agent 1 has the items  $a, b$ , and  $c$  as the most valuable ones.

Hence, in any of the two possible executions of Round-Robin (depending on the ordering of the agents), the resulting allocation will be  $A = (\{a, b, c\}, \{d, e, f\})$ . Notice that  $v_2(\{a, b, c, d, e, f\}) = 19$  while for every  $o \in \{a, b, c\}$ , it holds  $v_2(\{d, e, f, o\}) = 9 < v_2(\{a, b, c, d, e, f\})/2$ , implying that allocation  $A$  is not PROP1.  $\square$

## 5 Computing PROP1 Allocations for Satiating Subadditive Goods

In this section, we present our strongest algorithmic result, stated as follows.

**Theorem 11.** *There exists a polynomial-time algorithm that, on input any allocation instance with satiating subadditive goods, returns a PROP1 allocation.*

---

Algorithm 1: An algorithm producing a PROP1 allocation for allocation instances with satiating subadditive goods

---

**Require:** An allocation instance with  $n$  agents and a set of satiating subadditive goods  $M$

**Ensure:** A PROP1 allocation  $A$

```

1:  $A \leftarrow$  an arbitrary allocation
2:  $A \leftarrow$  improve( $A$ )
3: while  $P1(A) \neq [n]$  do
4:    $i \leftarrow$  an arbitrary agent in  $[n] \setminus P1(A)$ 
5:    $j \leftarrow$  an arbitrary agent in  $P(A)$ 
6:    $g \leftarrow$  an arbitrary item in  $A_j$ 
7:    $A_j \leftarrow A_j \setminus \{g\}$ ;  $A_i \leftarrow A_i \cup \{g\}$ 
8:    $A \leftarrow$  improve( $A$ )
9: end while
10: return  $A$ 

```

---

*Proof.* We will prove Theorem 11 using Algorithm 1. The input of the algorithm is an allocation instance with  $n$  agents and a set  $M$  of  $m$  satiating subadditive goods. The algorithm uses the three subroutines  $P(\cdot)$ ,  $P1(\cdot)$ , and  $\text{improve}(\cdot)$ .  $P(\cdot)$  takes as input an allocation and returns the set of agents that satisfy the proportionality conditions. Formally, an agent  $i$  belongs to set  $P(A)$  for an allocation  $A = (A_1, A_2, \dots, A_n)$  if  $n \cdot v_i(A_i) \geq v_i(M)$ . Similarly,  $P1(\cdot)$  takes as input an allocation and returns the set of agents that satisfy the PROP1 conditions. Formally, an agent  $i$  belongs to set  $P1(A)$  for an allocation  $A = (A_1, A_2, \dots, A_n)$  if it is either  $n \cdot v_i(A_i) \geq v_i(M)$  (i.e., if  $i \in P(A)$ ) or there exists an item  $g \notin A_i$  such that  $n \cdot v_i(A_i \cup \{g\}) \geq v_i(M)$ .

The subroutine  $\text{improve}(\cdot)$  takes as input an allocation  $A$  and works as follows. If there is an agent  $i \in [n]$  with  $i \in \arg \max_{j \in [n]} v_i(A_j)$ ,  $\text{improve}(A)$  returns  $A$ . Otherwise, it builds the directed graph  $G(A)$  containing a node corresponding to each agent  $i \in [n]$ . For each agent  $i \in [n]$ ,  $G(A)$  contains the directed edge  $(i, e_i)$  where  $e_i$  is an agent in  $[n]$  that agent  $i$  envies the most, i.e.,  $e_i \in \arg \max_{j \in [n] \setminus \{i\}} v_i(A_j)$ . Since each node of graph  $G(A)$  has out-degree 1,  $G(A)$  contains at least one cycle. The call of  $\text{improve}(A)$  identifies an arbitrary such cycle  $C = (c_1, \dots, c_k)$  with  $e_{c_k} = c_1$  and  $e_{c_t} = c_{t+1}$  for  $t = 1, 2, \dots, k-1$  and redistributes the bundles of  $A$  so that the agent (corresponding to node)  $c_k$  gets bundle  $A_{c_1}$  and agent  $c_t$  gets bundle  $A_{c_{t+1}}$  for  $t = 1, 2, \dots, k-1$ . By the definition of subroutine  $\text{improve}(\cdot)$ , we have the following property.

**Lemma 12.** *Let  $A$  be the allocation returned after an application of subroutine  $\text{improve}(\cdot)$ . Then,  $P(A) \neq \emptyset$ . Furthermore, every agent who was reassigned a bundle by subroutine  $\text{improve}(\cdot)$  belongs to set  $P(A)$ .*

*Proof.* Consider the application of subroutine  $\text{improve}(\cdot)$  on an allocation  $A$ . If it did not modify allocation  $A$ , this means that there is some agent  $i \in [n]$  such that  $v_i(A_i) \geq v_i(A_j)$  for every agent  $j \in [n]$ . Summing over these  $n$  inequalities and using the subadditivity of the valuation func-

tion  $v_i$ , we get

$$n \cdot v_i(A_i) \geq \sum_{j \in [n]} v_i(A_j) \geq v_i(M),$$

implying the proportionality condition for agent  $i$  and, hence,  $i \in P(A)$ .

If the application of subroutine  $\text{improve}(\cdot)$  modified allocation  $A$ , then a non-empty set of agents were reallocated their most valuable bundle. For each such agent  $i$ , their valuation after the redistribution of the bundles is  $v_i(A_i) \geq v_i(A_j)$  for every  $j \in [n]$  (here, we use  $A$  to denote the allocation obtained after the application of subroutine  $\text{improve}(\cdot)$ ). Using the same argument as in the previous paragraph, we get that all these agents satisfy the proportionality conditions and belong to  $P(A)$ .  $\square$

Algorithm 1 works as follows. It starts with an arbitrary allocation (line 1) on which subrouting  $\text{improve}(\cdot)$  is applied (line 2). Then, as long as there are agents for whom the PROP1 condition is not satisfied (i.e., the condition  $P1(A) \neq [n]$  in line 3), the algorithm runs the following loop. It identifies an arbitrary agent  $i$  for whom the PROP1 condition is not satisfied (line 4) and an arbitrary agent  $j$  for whom the proportionality condition is satisfied (line 5). Such agents do exist since the algorithm entered the while-loop and allocation  $A$  has been obtained after applying subroutine  $\text{improve}(\cdot)$ . Then, the algorithm picks an arbitrary item  $g$  from the bundle  $A_j$  (in line 6; again, such an item exists because agent  $j$  belongs to  $P(A)$  and, thus, satisfies the proportionality condition, i.e.,  $v_j(A_j) \geq v_j(M)/n > 0$ ). The current allocation is modified (in line 7) by moving item  $g$  from bundle  $A_j$  to bundle  $A_i$ , and subroutine  $\text{improve}(\cdot)$  is applied to this modified allocation (line 8). When leaving the while-loop, the algorithm returns the current allocation (line 10).

The above discussion guarantees that all steps in the while-loop are well-defined. Furthermore, if the algorithm exits the while-loop and terminates, it will clearly return a PROP1 allocation (since  $P1(A) = [n]$ , i.e., all agents would satisfy the PROP1 condition in this case). To complete the proof of correctness, we need to prove that the algorithm terminates given any allocation instance with satiating subadditive goods.

We use a potential function argument. Denote by  $A^0$  the allocation obtained after the execution of line 2 of Algorithm 1 and by  $A^t$  the allocation obtained after the  $t$ -th execution of line 8 (i.e., after the  $t$ -th execution of the while-loop). With some abuse in notation, we use  $P1(A^t)$  and  $P(A^t)$  to denote the set of agents for whom the PROP1 and proportionality conditions are satisfied in allocation  $A^t$ . The proof of the next lemma is omitted. It follows due to the fact that in each execution of the while-loop of Algorithm 1, an item from the bundle of an agent satisfying proportionality is given to an agent not satisfying PROP1; after the move, the first agent still satisfies PROP1.

**Lemma 13.** *For every integer  $t \geq 0$ ,  $P1(A^t) \subseteq P1(A^{t+1})$ .*

Lemma 13 proves the monotonicity of the quantity  $|P1(A^t)|$  in terms of number of executions  $t$  of the while-

loop of Algorithm 1. To show that the algorithm terminates, we furthermore need to show that  $|P1(A^t)|$  strictly increases after a polynomial number of executions of the while-loop. This is proved in the next lemma.

**Lemma 14.** *For every non-negative integer  $t$  with  $|P1(A^t)| < n$ , there is an integer  $t' \leq t + m$  such that  $|P1(A^{t'})| > |P1(A^t)|$ .*

*Proof.* For the sake of contradiction, let  $t$  be a non-negative integer such that  $|P1(A^t)| < n$  and assume that  $|P1(A^{t+m})| \leq |P1(A^t)|$ . By Lemma 13, this means that  $P1(A^{t+m}) = P1(A^t)$ . Hence, after the  $t$ -th execution and until after the  $(t+m)$ -th execution of the while-loop,  $m$  distinct items (i.e., all items) have moved to bundles of agents in  $[n] \setminus P1(A^{t+m})$ . Hence, the agents in set  $P1(A^{t+m})$  have no value at all in allocation  $A^{t+m}$ . Hence,  $v_k(A_k^{t+m}) < v_k(M)/n$ , for each agent  $k \in [n]$  and  $P(A^{t+m}) = \emptyset$ , which contradicts Lemma 12, as allocation  $A^{t+m}$  is obtained after running the subroutine `improve( $\cdot$ )`.  $\square$

Hence, Lemma 14 implies that every at most  $m$  executions of the while-loop, the number of agent satisfying the PROP1 conditions increases by at least 1. This means that Algorithm 1 computes a PROP1 allocation after at most  $n \cdot m$  executions of the while-loop. The proof of Theorem 11 is now complete.  $\square$

The proof of Theorem 11 essentially shows that Algorithm 1 computes a PROP1 allocation for satiating subadditive goods after  $O(n \cdot m)$  item allocations or reallocations. For allocation instances with satiating submodular goods, we can compute a PROP1 allocation much faster by combining Round-Robin and Algorithm 1. The main change is to replace line 1 of Algorithm 1 by an execution of Round-Robin until its second-to-last round. By Theorem 7, the partial allocation computed by Round-Robin after the second-to-last round satisfies PROP1 for each agent. We can complete this allocation by assigning all items left for the last Round-Robin round to a single arbitrary agent; this complete allocation satisfies PROP1 for all agents but one. By running Algorithm 1 using this initial allocation, we have  $|P1(A^0)| \geq n - 1$  initially, and, using Lemma 14, we get that all agents will satisfy PROP1 after at most  $m$  executions of the while-loop of Algorithm 1. The following statement summarizes this discussion.

**Theorem 15.** *There exists a polynomial-time algorithm which, given an allocation instance with  $m$  satiating submodular goods, returns a PROP1 allocation after at most  $O(m)$  item allocations and reallocations.*

## 6 Can PROP1 Allocations Be Pareto-Optimal?

We now explore the compatibility of PROP1 and Pareto-optimality. Following Caragiannis et al. (2019), we extend the definition of the maximum Nash welfare allocation to capture instances in which no allocation yields positive valuation to all agents for their bundles. So, the maximum Nash welfare allocation is one that maximizes the number of

agents with non-zero valuation for their bundle and, under this condition, it maximizes the product of non-zero agent valuations.

Our first result showcases another facet of the “unreasonable fairness” of allocations with maximum Nash welfare.

**Theorem 16.** *In any allocation instance with monotone submodular goods, the maximum Nash welfare allocation is PROP1.*

We have two proofs of Theorem 16 (both are omitted). The first one is direct and long. A shorter proof follows by first using a result of Caragiannis et al. (2019) stating that maximum Nash welfare allocations have a fairness property called *marginal envy-freeness up to some good* (MEF1) and then proving that MEF1 allocations are also PROP1.

As a maximum Nash welfare allocation is Pareto-optimal, we obtain the following corollary.

**Corollary 17.** *In any allocation instance with monotone submodular goods, a PROP1 and Pareto-optimal allocation exists.*

Unfortunately, PROP1 can be incompatible with Pareto-optimality if slightly more general (i.e., monotone XOS) valuation functions are allowed.

**Theorem 18.** *There exists an allocation instance with monotone XOS goods, in which no Pareto-optimal allocation is PROP1.*

*Proof.* We use the allocation instance with three agents having the same XOS valuation function for seven items that we used in the proof of Theorem 5. The only PROP1 allocations should give three items to some agent and two items to each of the other two. Indeed, if an agent got at most one item, the valuation for any superset including an extra item would be 2 while the proportionality threshold is  $7/3$ . Then, the (non-PROP1) allocation that gives five items to the agent who initially got the three items and one item to each of the other two agents is a Pareto improvement; the first agent has strictly better value and the other two are not worse off.  $\square$

## 7 Conclusion and Open Problems

We have presented new results regarding the existence and efficient computation of PROP1 allocations for instances with non-additive (i.e., submodular and subadditive) and possibly non-monotone valuation functions over goods. Our work leaves several open problems, including the following. First, is PROP1 and Pareto-optimality compatible on allocation instances with satiating submodular goods? Second, is there a polynomial-time algorithm for computing a PROP1 and Pareto-optimal allocation for monotone submodular goods? Third, what is the price of fairness with respect to Nash welfare of PROP1 allocations in subadditive instances? The proof of Theorem 18 can be modified to yield a price of fairness lower bound higher than 1. Finally, for classes of allocation instances in which PROP1 is not compatible with Pareto-optimality, what is the best approximation of PROP1 that is compatible with Pareto-optimality? Can such approximately PROP1 and Pareto-optimal allocations be computed efficiently?

## Acknowledgements

This work has been partially supported by Independent Research Fund Denmark (DRF) under grant 2032-00185B.

## References

- Amanatidis, G.; Aziz, H.; Birmpas, G.; Filos-Ratsikas, A.; Li, B.; Moulin, H.; Voudouris, A. A.; and Wu, X. 2023a. Fair division of indivisible goods: Recent progress and open questions. *Artificial Intelligence*, 322: 103965.
- Amanatidis, G.; Birmpas, G.; Lazos, P.; Leonardi, S.; and Reiffenhäuser, R. 2023b. Round-robin beyond additive agents: Existence and fairness of approximate equilibria. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC)*, 67–87.
- Aziz, H.; Caragiannis, I.; Igarashi, A.; and Walsh, T. 2022. Fair allocation of indivisible goods and chores. *Autonomous Agents & Multi Agent Systems*, 36(1): 3.
- Aziz, H.; Moulin, H.; and Sandomirskiy, F. 2020. A polynomial-time algorithm for computing a Pareto optimal and almost proportional allocation. *Operations Research Letters*, 48(5): 573–578.
- Barman, S.; and Krishnamurthy, S. K. 2019. On the proximity of markets with integral equilibria. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, 1748–1755.
- Barman, S.; and Krishnamurthy, S. K. 2020. Approximation algorithms for maximin fair division. *ACM Transactions on Economics and Computation*, 8(1): 5:1–5:28.
- Barman, S.; and Suzuki, M. 2024. Compatibility of fairness and Nash welfare under subadditive valuations. *CoRR*, abs/2407.12461.
- Barman, S.; and Verma, P. 2025. Fair division beyond monotone valuations. *CoRR*, abs/2501.14609.
- Budish, E. 2011. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6): 1061–1103.
- Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A. D.; Shah, N.; and Wang, J. 2019. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation*, 7(3): 12:1–12:32.
- Chaudhury, B. R.; Garg, J.; and Mehta, R. 2021. Fair and efficient allocations under subadditive valuations. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 5269–5276.
- Conitzer, V.; Freeman, R.; and Shah, N. 2017. Fair public decision making. In *Proceedings of the 18th ACM Conference on Economics and Computation (EC)*, 629–646.
- Dobzinski, S.; Li, W.; Rubinfeld, A.; and Vondrák, J. 2024. A constant-factor approximation for Nash social welfare with subadditive valuations. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC)*, 467–478.
- Feige, U. 2009. On maximizing welfare when utility functions are subadditive. *SIAM Journal on Computing*, 39(1): 122–142.
- Feige, U.; Mirrokni, V. S.; and Vondrák, J. 2011. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4): 1133–1153.
- Feldman, M.; Mauras, S.; and Ponitka, T. 2024. On optimal tradeoffs between EFX and Nash welfare. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, 9688–9695.
- Garg, J.; and Sharma, E. 2025. Exploring relations among fairness notions in discrete fair division. *CoRR*, abs/2502.02815.
- Krause, A.; and Golovin, D. 2014. Submodular function maximization. In Bordeaux, L.; Hamadi, Y.; and Kohli, P., eds., *Tractability: Practical Approaches to Hard Problems*, 71–104. Cambridge University Press.
- Lehmann, B.; Lehmann, D.; and Nisan, N. 2006. Combinatorial auctions with decreasing marginal utilities. *Games & Economic Behavior*, 55(2): 270–296.
- Lipton, R. J.; Markakis, E.; Mossel, E.; and Saberi, A. 2004. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, 125–131.
- McGlaughlin, P.; and Garg, J. 2020. Improving Nash social welfare approximations. *Journal of Artificial Intelligence Research*, 68: 225–245.
- Seddighin, M.; and Seddighin, S. 2025. Beating the logarithmic barrier for the subadditive maximin share problem. In *Proceedings of the 26th ACM Conference on Economics and Computation (EC)*, 764–782.
- Steinhaus, H. 1948. The problem of fair division. *Econometrica*, 16: 101–104.