Algorithmically Fair Maximization of Multiple Submodular Objective Functions

Georgios Amanatidis
Athens University of Economics and Business
Archimedes / Athena RC
Athens, Greece
g.d.amanatidis@aueb.gr

Georgios Birmpas
University of Liverpool
Liverpool, United Kingdom
gebirbas@gmail.com

Philip Lazos
Jump Trading
London, United Kingdom
plazos@gmail.com

Stefano Leonardi
Sapienza University of Rome
Rome, Italy
leonardi@diag.uniroma1.it

Rebecca Reiffenhäuser
University of Amsterdam
Amsterdam, The Netherlands
r.e.m.reiffenhauser@uva.nl

ABSTRACT

Constrained maximization of submodular functions poses a central problem in combinatorial optimization. In many realistic scenarios, a number of agents need to maximize multiple submodular objectives over the same ground set. We study such a setting, where the different solutions must be disjoint, and thus, questions of algorithmic fairness arise. Inspired from the fair division literature, we suggest a simple round-robin protocol, where agents are allowed to build their solutions one item at a time by taking turns. Unlike what is typical in fair division, however, the prime goal here is to provide a fair algorithmic environment; each agent is allowed to use any algorithm for constructing their respective solutions. We show that just by following simple greedy policies, agents have solid guarantees for both monotone and non-monotone objectives, and for combinatorial constraints as general as p-systems (which capture cardinality and matroid intersection constraints). In the monotone case, our results include the first approximate EF1-type guarantees under such general constraints. Further, although following a greedy policy may not be generally optimal, we show that consistently performing better than that is computationally hard.

KEYWORDS

Submodular maximization; round-robin; fairness; fair division; EF1

ACM Reference Format:
Georgios Amanatidis, Georgios Birmpas, Philip Lazos, Stefano Leonardi, and Rebecca Reiffenhäuser. 2025. Algorithmically Fair Maximization of Multiple Submodular Objective Functions. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Dealing with competing interests of selfish agents poses the main challenge at the heart of many directions of research, where we need to manage the agents' private agendas in pursuit of a different, global goal. For instance, in *auction design*, one might aim to elicit honest bidding to reach a welfare-maximizing assignment of items; in *social choice*, typically the goal is to map individual preferences to outcomes that serve society as a whole; in *fair division*, one attempts to serve the agents' interests in a way that satisfies some notion of equal treatment. We consider a natural setting that shares a similar flavor: there is a number of agents, each of whom aims to obtain a subset of discrete resources that maximizing their objective function. Of course, in general, the agents' objective functions cannot all be maximized at the same time, as they compete over some of the common resources. What differentiates our setting is that we do not want to explicitly solve a complicated combinatorial optimization problem on behalf of the agents.

As an illustrating example, consider the problem of influence maximization, e.g., as defined in Breuer et al. [13]. The goal here is to pick a number of vertices, such as popular users on a social network graph, which maximizes the influence on the rest of the users, assuming that users are independently influenced by their neighbors. Now consider a network (e.g., Facebook) on which multiple companies want to advertise their competing products (i.e., solve an influence maximization problem each, on disjoint sets of nodes). How could one implement a solution that all parties agree is "algorithmically fair"? What is more, how should one formalize and exploit the fact that, in large-scale instances like this, the optimal value of an agent is not expected to be particularly affected by the controlled removal of resources? Typically, it is taken for granted that the central authority of the (potentially vast) network is willing to collect all the relevant information from their clients and then micromanage the outcome of every single such transaction.

Here we assume that the role of such a central authority is not that of deciding everyone's final outcome, but rather of setting an instance-independent algorithmic environment that gives to all the agents a fair chance to attain their goal. That is, all agents are required to adhere to a certain *protocol* while they try to maximize their objective functions. Specifically, inspired from the fair division literature, we investigate the probably most intuitive protocol of having agents take turns, picking resources in a *round-robin* fashion.

So, contrary to classic approaches in social choice or auctions, we never collect the valuations, nor are we interested in eliciting truthful reports. Instead we opt for the much simpler solution of

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

defining a selection protocol and then show that it is possible for every agent to have a guarantee on the quality of their final solution, even by following a simple policy. Note that we do not imply any specific way this should be done; whether the agents indeed realize their potential, or what algorithms are actually used in the background, is of no direct interest. Rather, the aim here is to provide the participants with an agreeable framework that reasonably limits the negative effects of competition and avoids a possible *winner-takes-all* situation. This approach has the additional advantage of delegating any computational task to the agents rather than the protocol itself, making it both easy to implement and transparent / explainable to its users.

A further advantage of our approach is that there is an easy way for an agent to get a reasonably good solution. Assuming there are n agents, each with an objective function f_i over a set of items M, note that no algorithmic framework for the problem can guarantee an approximation factor of more than $1/n$ to all the agents with respect to the optimal value they could obtain without competition, even when randomization is allowed. This is straightforward to see in the example where $f_i(S) = |S|$, for all $S \subseteq M$ and $i \in [n]$. The best worst-case guarantee for the least satisfied agent is $1/n$ and, thus, we would like an algorithmic framework with comparable guarantees for highly competitive instances like this, which ideally can significantly improve for "nicer" instances. What we show is that by following a simple greedy policy throughout our Round-Robin protocol, an agent can achieve such guarantees, even when they have strong feasibility constraints. And while we are not interested in enforcing any specific policy, we show that improving over these greedy policies is NP-hard for the agents, even in the very simplest of settings. Interestingly, as one moves towards instances that are more robust to competition—under a *large market* flavored assumption we call (α, β)-robustness (Definition 2.4)—the guarantees of the protocol improve up to the point where constant approximation guarantees can be achieved for everyone.

There are obvious connections of our problem with fair division, even in the choice of the protocol itself. Fully exploring these connections or using fairness criteria as our benchmarks is not our primary goal here. Nevertheless, there are implications of our results on fair division which we do discuss in Section 4.1.

Our Contribution. We initiate the study of *coordinated maximization protocols* for the problem where n agents want to maximize a submodular function each, say f_i for agent i, subject to a combinatorial constraint; all the functions are defined over a common ground set M and the agents' solutions should be disjoint. We suggest and analyze a natural Round-Robin protocol (Protocol 1), as well as a randomized variant of it (Protocol 2), that allow the agents to take turns in adding at most one item from the ground set to their solution sets at a time. In particular, we show that:

- By employing a simple greedy policy (Policy 1), an agent i with a monotone submodular objective and a p_i-system constraint can achieve a good approximation of the optimal solution that is still available when they first get to choose an item, OPT_i^-. In particular, they achieve a value of at least $\text{OPT}_i^-/(n + p_i)$ (Theorem 4.1) or OPT_i^-/n for a cardinality constraint (Theorem 4.3). As one moves to "nicer" instances that are less affected by competition (see Definition 2.4), these guarantees gracefully improve beyond $\text{OPT}_i/(p_i+2)$ or $\text{OPT}_i/3$, respectively (Theorem 4.10 and Corollary 4.11), which is almost what is possible in polynomial time, even without any competition!
- Moreover, if agent i follows Policy 1, then the value they secure (disregarding any items picked before they get their first turn) is at least $1/(p_i + 2)$ times the value for any feasible subset of any other agent's solution (Theorem 4.4). An immediate implication of this, is the first EF1-type result for settings with so strong constraints on the allocation (Corollary 4.8).
- An agent i with a *non-monotone* submodular objective and a p_i-system constraint can still achieve a value of at least $\text{OPT}_i^-/(4n+4p_i+2)$ (or $\text{OPT}_i^-/(4n+2)$ for a cardinality constraint) (Theorem 6.1) by simultaneously building *two* greedy solutions (Policy 2). Like in the monotone case, as we move to instances that are more robust to competition, these guarantees gradually improve beyond $\text{OPT}_i/(4p_i + 4)$ or $\text{OPT}_i/8$, respectively (Theorem 6.2 and Corollary 6.3), which is within a constant factor from the best possible guarantee even with a single agent!
- Fix some $j \in [n]$ and any $\varepsilon \in (0, 0.3)$. Even when all agents in $[n] \setminus \{j\}$ have additive objective functions and follow a greedy policy, there is no polynomial-time algorithm that can improve over Policy 1 by a factor of $(1 + \varepsilon)$ whenever this is possible, unless P = NP (Theorem 5.1).
- When randomness is employed in the most natural way, all the aforementioned worst-case guarantees for an agent i translate into comparable ex-ante guarantees with respect to OPT_i instead of OPT_i^- (Theorem 7.1). Further, the assumptions needed to obtain $O(p_i)$ approximation guarantees with respect to OPT_i, are significantly weaker (Theorem 7.2).

In short, the Round-Robin protocols allow agents with strong combinatorial constraints to obtain a $1/\Theta(n)$ fraction of the optimal value available when they enter the process and, in expectation, of their optimal value overall. This remains true even when moving to non-monotone objective functions and improves to $1/\Theta(p_i)$ for instances that are robust to competition, which is asymptotically best-possible in polynomial time. The agents can achieve these guarantees by employing extremely simple greedy strategies, and going beyond those, while possible, is generally computationally hard even in very simple cases. Finally, no agent with a monotone objective who chooses greedily will value someone else's set much higher than their own, establishing a formal notion of algorithmically fair treatment.

Note that none of our results follows in any obvious way from known results in constrained submodular maximization and that naively applying standard analyses would result in approximation factors of order $\Theta(np_i)$ rather than $\Theta(n + p_i)$. Our careful analysis of greedy policies is combined with novel mappings that associate items added in the solution of an agent with items "lost" for that agent (due to myopic choices or other agents) in multiple sets at once; see Lemma 4.2.

Related Work. There is a vast literature on optimizing a submodular function, with or without constraints, dating back to the seminal works of Nemhauser et al. [32] and Fisher et al. [22]. For an overview of the main variants of the problem, we refer the interested reader to the survey of Buchbinder and Feldman [14] and the references therein. For maximizing a monotone submodular function subject to a p-system constraint, in particular, the simple greedy algorithm that adds to the solution the item with the highest marginal value in each step achieves an approximation factor of $1/(p+1)$ [22], which improves to $1-1/e$ for a cardinality constraint [32]. The latter is the best possible approximation factor, assuming that P $\neq$ NP [20]. Moreover, Badanidiyuru and Vondrák [9] showed that even maximizing an *additive* function subject to a p-system constraint, within a factor of $1/(p-\varepsilon)$ for any fixed $\varepsilon > 0$, requires exponentially many independence oracle queries.

The state-of-the-art for non monotone objective functions subject to a p-system constraint is much more recent. Gupta et al. [25] introduced a *repeated greedy* framework, which can achieve an approximation factor of $1/2p$, as shown by Mirzasoleiman et al. [31]. The best known factor is $1/(p+\sqrt{p})$ by Feldman et al. [21], using the *simultaneous greedy* framework. Simultaneous greedy algorithms bypass non-monotonicity by constructing *multiple* greedy solutions *at the same time*. This idea was first introduced by Amanatidis et al. [5] for a knapsack constraint and a p-system constraint in a mechanism design setting, and has been used successfully in a number of variants of the problem since [18, 26, 27, 33, 35]. The impressive versatility of this approach, however, became apparent in the recent work of Feldman et al. [21], who fully develop a framework that achieves the best known factors for a p-system constraint and multiple knapsack constraints combined.

The very simple Round-Robin algorithm that we use as the basis for our Protocols 1 and 2 is a fundamental procedure encountered throughout the fair division literature [8, 16, 30], often modified [e.g., 7] or as a subroutine of more complex algorithms [e.g., 6, 28]. For a discussion on the connection of our work with fair division and some further related work, see Section 4.1.

2 PRELIMINARIES

Before we formally state the problem, we introduce some notation and give the relevant definitions and some basic facts. Let $M = [m] = \{1, 2, ..., m\}$ be a set of m items. For a function $f : 2^M \to \mathbb{R}$ and any sets $S \subseteq T \subseteq M$ we use the shortcut $f(T \mid S)$ for the *marginal value of T with respect to S*, i.e., $f(T \mid S) = f(T \cup S) - f(S)$. If $T = \{i\}$ we simply write $f(i \mid S)$.

Definition 2.1. A function f is *submodular* if and only if $f(i \mid S) \geq f(i \mid T)$ for all $S \subseteq T \subseteq M$ and $i \notin T$.

If f is non-decreasing, i.e., if $f(S) \leq f(T)$ for any $S \subseteq T \subseteq M$, we just refer to it as being *monotone* in this context. We consider normalized (i.e., $f(\emptyset) = 0$), non-negative submodular objective functions, both monotone and non-monotone.

Our algorithmic goal is to maximize multiple submodular functions at the same time, each one with its own constraint, through simple protocols. As we imply in the Introduction, the term *protocol* here refers to a procedure that (a) does not take any general input about the valuation functions or the constraints of the agents but is, possibly, allowed to ask a limited number of simple queries, (b) performs little to no computation itself, and (c) chooses one agent at a time and allows them to add a number of items to their solution, possibly from a subset of the available items. The constraints can be as general as p-system constraints.

Definition 2.2. Given a set M, an *independence system* for M is a family $\mathcal{I}$ of subsets of M, whose members are called the *independent sets* of M and satisfy *(i)* $\emptyset \in \mathcal{I}$, and *(ii)* if $B \in \mathcal{I}$ and $A \subseteq B$, then $A \in \mathcal{I}$. We call $\mathcal{I}$ a *matroid* if it is an independence system and it also satisfies the exchange property *(iii)* if $A, B \in \mathcal{I}$ and $|A| < |B|$, then there exists $x \in B \setminus A$ such that $A \cup \{x\} \in \mathcal{I}$.

Given a set $S \subseteq M$, a maximal independent set contained in S is called a *basis* of S. The upper rank $\mathrm{ur}(S)$ (resp. lower rank $\mathrm{lr}(S)$) is defined as the largest (resp. smallest) cardinality of a basis of S.

Definition 2.3. A *p-system* for M is an independence system for M, such that $\max_{S \subseteq M} \mathrm{ur}(S)/\mathrm{lr}(S) \leq p$.

Many combinatorial constraints are special cases of p-systems for small values of p. A *cardinality* constraint, i.e., feasible solutions contain up to a certain number of items, induces a 1-system. A *matroid* constraint, i.e., feasible solutions belong to a given matroid, also induces a 1-system; in fact, a cardinality constraint is a special case of a matroid constraint. More generally, constraints imposed by the intersection of k matroids, i.e., feasible solutions belong to the intersection of k given matroids, induce a k-system; matching constraints are examples of such constraints for $k = 2$.

Constrained Maximization of Multiple Submodular Objectives (for short, MultiSubmod): For $i \in [n]$, let $f_i : 2^M \to \mathbb{R}$ be a submodular function and $\mathcal{I}_i \subseteq 2^M$ be a p_i-system. Find disjoint subsets of M, say $S_1, \ldots, S_n$, such that $S_i \in \mathcal{I}_i$ and $f_i(S_i) = \max_{S \in \mathcal{I}_i} f_i(S) := \mathrm{OPT}_i$.

We think of f_i and $\mathcal{I}_i$ as being associated with an agent $i \in [n]$, i.e., they are i's objective function and combinatorial constraint, respectively. Of course, maximizing all the functions at once may be impossible as these objectives could be competing with each other. Naturally, we aim for approximate solutions, i.e., for $S_1, \ldots, S_n$, such that $f_i(S_i) \geq \rho\, \mathrm{OPT}_i$, for all $i \in [n]$ and a common approximation ratio ρ (possibly a function of n). As a necessary compromise in our setting, we often use OPT_i^- instead of OPT_i as the benchmark for agent i in the worst-case. We will revisit and formalize this benchmark in Section 4 but, essentially, if some items have already been allocated right before anything is added to S_i for the first time, then OPT_i^- is the value of an optimal solution for i still remaining available at that time.

We also intend to evaluate our protocols on instances that are more robust to competition. We want to capture the behavior one would expect in large-scale applications of our setting, e.g., in our running example of multiple firms competing to maximize their influence on a vast social network. That is, the value of an optimal solution of an agent should not be greatly affected by the removal of a reasonably sized subset of items. This is formalizeed below.

Definition 2.4. Let $\alpha \in \mathbb{N}, \beta \in \mathbb{R}_+$. An instance of MultiSubmod is *(α, β)-robust with respect to agent i* if there are α disjoint sets $O_{i1}, \ldots, O_{i\alpha} \subseteq M$, so that $O_{ij} \in \mathcal{I}_i$ and $\beta \cdot f_i(O_{ij}) \geq \mathrm{OPT}_i$, for $j \in [\alpha]$.

That is, if an instance is (α, β)-robust with respect to agent i, then it contains at least α independent solutions of value within a factor of β from i's optimal value. Clearly, any instance is $(1, 1)$-robust with respect to any agent. When we refer to instances that are more robust to competition, we essentially mean instances that are $(\Omega(n), O(1))$-robust with respect to everyone.

Besides the definition of submodularity given above, there are alternative equivalent definitions that will be useful later. These are summarized in the following result of Nemhauser et al. [32].

Theorem 2.5 (Nemhauser et al. [32]). *A function $f : 2^M \to \mathbb{R}$ is submodular if and only if, for all $S, T \subseteq M$, $f(T) \leq f(S) + \sum_{i \in T \setminus S} f(i \mid S) - \sum_{i \in S \setminus T} f(i \mid S \cup T \setminus \{i\})$. Further, f is monotone submodular if and only if, for all $S, T \subseteq M$, $f(T) \leq f(S) + \sum_{i \in T \setminus S} f(i \mid S)$.*

As it is common in the submodular optimization literature, we assume oracle access to the functions via value queries, i.e., for $i \in [n]$, we assume the existence of a polynomial-time value oracle that returns $f_i(S)$ when given as input a set S. Similarly, we assume the existence of independence oracles for the constraints, i.e., for $i \in [n]$, we assume there is a polynomial-time algorithm that, given as input a set S, decides whether $S \in \mathcal{I}_i$ or not.

3 A ROUND-ROBIN FRAMEWORK

We present our simple protocol, which is aligned with how the majority of submodular maximization algorithms work (i.e., building one or more solutions one item at a time): here, the agents take turns according to a fixed ordering and in each step the active agent chooses (at most) one available item to add to their solution. Note that we do not impose how this should be done; it is the agents' task to decide how an item will be chosen, whether their solution should remain feasible throughout the protocol or they maintain a feasible solution within a larger chosen set, etc. We stress again that this approach has the significant advantage of delegating any computationally challenging task to the agents.

Note that when we refer to the *policy* of an agent, we mean their overall algorithmic strategy; how they make their algorithmic choices, given full information about other agents' objective functions, constraints, and current solutions. So when we write $\mathcal{A}_i(S_i\,; Q)$ in line 4 of the description of Protocol 1, in general, $\mathcal{A}_i(\cdot)$ can be a function of all that information. Later, in Section 5, when we make the distinction and talk about the *algorithm* of an agent, we typically consider other agents' objective functions and policies fixed. We do not formalize this further, as the main policies we consider in this work are independent of any information about other agents.

We could have any ordering fixed by Protocol 1 at its very beginning, i.e., a permutation $s_1, \dots, s_n$ of $[n]$, such that s_i is the i-th agent to choose their first item. To simplify the presentation, we assume that $s_i = i$, for all $i \in [n]$. This is without loss of generality, as it only involves a renaming of the agents before the main part of the protocol begins. We revisit this convention in Section 7 where we randomize over all possible agent permutations.

Protocol 1 Round-Robin($\mathcal{A}_1, \dots, \mathcal{A}_n$)
(For $i \in [n]$, $\mathcal{A}_i$ is the policy of agent i.)

```
1: Q = M; k = ⌈m/n⌉
2: for r = 1, ..., k do
3:     for i = 1, ..., n do
4:         j = A_i(S_i ; Q)        (where j could be a dummy item)
5:         Q = Q \ {j}
```

Refining Our Benchmark. Note that even having a guarantee of OPT_i/n, for all $i \in [n]$, is not always possible (even approximately) in the worst case. Indeed, let us modify the example from the introduction so that there are a few very valuable items: for all $i \in [n]$, let $f_i(S \cup T) = |S| + L|T|$, for any $S \subseteq M_1$ and $T \subseteq M_2$, where $M = M_1 \cup M_2$ and $L \gg |M|$. When $|M_2| < n$, then the best possible value for the least satisfied agent is a $1/(n - |M_2|)$ fraction *not of* OPT_i *but of i's optimal value in a reduced instance where the items in M_2 are already gone.*

With this in mind, we are going to relax the OPT_i benchmark a little. As mentioned in Section 2, we define OPT_i^- to be the value of an optimal solution available to agent i, given that $i - 1$ items have been lost before i gets to pick their first item. Note that, for $i \in [n]$,

$$\mathrm{OPT}_i^- \geq \min_{M' \in \binom{M}{m-i+1}} \max_{S \in \mathcal{I}_i | M'} f_i(S)\,,$$

i.e., OPT_i^- is always at least as large as the pessimistic prediction that agents before i will make the worst possible choices for i. The notation $\binom{M}{x}$ used here denotes the set of subsets of M of cardinality x and $\mathcal{I}|A$ denotes (the independence system induced by) the restriction of $\mathcal{I}$ on A, i.e., $\mathcal{I}|A = \{X \cap A : X \in \mathcal{I}\}$. By inspecting Definition 2.2, it is easy to see that the restriction of a p-system is a p-system.

It is worth mentioning that, while OPT_i^- is defined with respect to Protocol 1 here, its essence *is not an artifact* of how Round-Robin works. No matter which sequential protocol (or even algorithm with full information) one uses, if there are n agents, then it is unavoidable that someone will get their first item after r items are already gone, for any $r < n$; this is inherent to any setting with agents competing for resources.

In the next sections, we show that very simple greedy policies in Protocol 1 can guarantee value of at least $\mathrm{OPT}_i^- / \Theta(n + p_i)$ to any agent $i \in [n]$ who follows them. This fact implies novel results in constrained fair division but also allows for a randomized protocol where the corresponding ex-ante worst-case guarantees are in terms of OPT_i instead. Furthermore, is allows us to go beyond worst-case analysis and obtain much stronger guarantees for $(\Omega(n), O(1))$-robust instances that are almost best-possible in polynomial time.

4 THE EFFECTIVENESS OF GREEDY POLICIES FOR MONOTONE OBJECTIVES

We first turn to monotone submodular objective functions; we deal with non-monotonicity in Section 6. Simple greedy algorithms, where in every step a feasible item of maximum marginal utility is added to the current solution, have found extreme success in a wide range of submodular maximization problems. Hence, it is only natural to consider the following question:

> *What value can an agent guarantee for themselves if they always choose greedily?*

Next we show that an agent i can achieve strong bounds with respect to OPT_i^-. 'Strong' here refers to the fact that both Theorems 4.1 and 4.3 achieve constant factor approximations to OPT_i^- for constant n, but also that, for $n = 1$, Theorem 4.1 retrieves the best-known guarantee of the greedy algorithm for the standard algorithmic problem [22], which is almost best-possible for polynomially many queries [9]. Further, for (α, β)-robust instances these guarantees improve the larger α becomes. In particular, if an instance is $(\Omega(n), O(1))$-robust with respect to agent i, then i achieves a $1/\Theta(p_i)$ fraction of their *optimal* value OPT_i instead, which is asymptotically best-possible.

Formally, when we say that agent i *chooses greedily* we mean they choose according to the policy $\mathcal{G}_i$ below.

Policy 1 Greedy policy $\mathcal{G}_i(S_i\,;Q)$ of agent i. (S_i : current solution of agent i (initially $S_i = \emptyset$) ; Q : current set of available items)

$A = \{x \in Q \ : \ S_i \cup \{x\} \in \mathcal{I}_i\}$
if $A \neq \emptyset$ **then**
  $S_i = S_i \cup \{j\}$, where $j \in \arg\max_{z \in A} f(z \,|\, S_i)$
  return j
else
  return a dummy item (i.e., return nothing)

Theorem 4.1. *Any agent i with a p_i-system constraint, who chooses greedily in the Round-Robin protocol, builds a solution S_i such that $f_i(S_i) \geq \mathrm{OPT}_i^-/(n+p_i)$.*

Proof. Let O_i^- be an optimal solution for agent i on the set M_i of items still available after $i-1$ steps, i.e., right before agent i chooses their very first item; by definition, $\mathrm{OPT}_i^- = f_i(O_i^-)$. From i's perspective, it makes sense to consider the k-th round to last from when they get their k-th item until right before they choose their $(k+1)$-th item. We rename the items of M_i accordingly as $x_1^i, x_1^{i+1}, \ldots, x_1^{i-1}, x_2^i, \ldots$, i.e., item x_j^ℓ is the j-th item that agent ℓ chooses from the moment when agent i is about to start choosing; any items not picked by anyone (due to feasibility constraints) are arbitrarily added to the end of the list. Also, let $S_i^{(r)}$ denote the solution of agent i right before item x_r^i is added to it. Finally, set $s := |S_i|$. We are going to need the existence of a mapping $\delta : O_i^- \setminus S_i \to S_i$ with "nice" properties, as described in the following lemma for $a = 1$ and $Q_{i1} = O_i^-$; the general form of Lemma 4.2 is needed for the proof of Theorem 4.10.

Lemma 4.2. *Let $Q_{i1}, \ldots, Q_{ia} \in \mathcal{I}_i$ be disjoint feasible sets for agent i. There is a mapping $\delta : \bigcup_{j\in[a]} Q_{ij} \setminus S_i \to S_i$ with the following two properties, for all $x_r^i \in S_i, x \in \bigcup_{j\in[a]} Q_{ij} \setminus S_i$:*

(1) if $\delta(x) = x_r^i$, then $f_i(x \,|\, S_i^{(r)}) \leq f_i(x_r^i \,|\, S_i^{(r)})$, i.e., x is not as attractive as x_r^i when the latter is chosen;

(2) $|\delta^{-1}(x_r^i)| \leq n + ap_i - 1$, i.e., at most $n + ap_i - 1$ items of $\bigcup_{j\in[a]} Q_{ij} \setminus S_i$ are mapped to each item of S_i.

For now, we assume the lemma and apply the second part of Theorem 2.5 for O_i^- and S_i:

$$\begin{aligned}
f_i(O_i^-) &\leq f_i(S_i) + \sum_{x \in O_i^- \setminus S_i} f_i(x \,|\, S_i) \\
&\leq f_i(S_i) + \sum_{r=1}^{s} \sum_{x \in \delta^{-1}(x_r^i)} f_i(x \,|\, S_i) \\
&\leq f_i(S_i) + \sum_{r=1}^{s} \sum_{x \in \delta^{-1}(x_r^i)} f_i(x \,|\, S_i^{(r)}) \\
&\leq f_i(S_i) \quad + \sum_{r=1}^{s} (n + p_i - 1) f_i(x_r^i \,|\, S_i^{(r)}) \\
&= f_i(S_i) + (n + p_i - 1) f_i(S_i) = (n + p_i) f_i(S_i)\,.
\end{aligned}$$

where the second inequality follows from observing that $O_i^- \setminus S_i = \bigcup_{r=1}^{s} \delta^{-1}(x_r^i)$, the third follows from submodularity, and the fourth from Lemma 4.2. We conclude that $f_i(S_i) \geq \mathrm{OPT}_i^-/(n+p_i)$. □

Here one has to deal with the fact that an agent may miss "good" items not only by their own choices (that make such items infeasible), but also because others take such items in a potentially adversarial way. This complication calls for very careful mapping in Lemma 4.2 *which concentrates most of the technical difficulty of the proof* of Theorem 4.1. Note that the lemma allows for a mapping from the union of multiple disjoint feasible subsets, not just O_i^-. This comes handy for proving Theorem 4.10 but adds an extra layer of complexity in keeping track of different kinds of items in its proof. The proof is deferred to the full version of our paper [3] (as is the case for any other missing proof), but we give here the general idea, at least for the special case of a single set, O_i^-, as it is used for Theorem 4.1. An agent i may miss items from their optimal solution O_i^- for two reasons: (i) because of greedy choices that make them infeasible, and (ii) because other agents take them. It is easy to see that between any two greedy choices of i, there are at most $n-1$ such lost items of type (ii) and all have marginals that are no better than the marginal of the former choice. The next step is to show that after the first ℓ greedy choices, for any ℓ, there are at most ℓp_i lost items of type (i). Finally, we argue that this allows us to map at most p_i of them to each greedy choice of i so that they never have a larger marginal value than that greedy choice.

Specifically for cardinality constraints (which are 1-system constraints and include the unrestricted case when the cardinality constraint is at least $\lceil m/n \rceil$), we can get a slightly stronger version of Theorem 4.1. Note that Theorem 4.3 assumes that $n \geq 2$, and thus it does not contradict the inapproximability result of Feige [20] for the case of a single objective function. Here the analysis of the greedy policy for Protocol 1 is tight for any i, as follows by the example from the introduction (e.g., for $m = n$).

Theorem 4.3. *For $n \geq 2$, any agent i with a cardinality constraint, who chooses greedily in the Round-Robin protocol, builds a solution S_i such that $f_i(S_i) \geq \mathrm{OPT}_i^-/n$.*

A reasonable question at this point is whether this is a strong benchmark in general. The answer is both yes and no! On one hand, in Section 4.8 below, we get an EF1-type guarantee for any agent who chooses greedily, no matter what other agents do. On the other hand, there are instances where an agent can improve their solution by a factor of order $\Omega(n)$ by choosing items more carefully and taking other agents' objective functions into consideration (see also [36]). Yet, in Section 5 we show that it is NP-hard to consistently improve on Policy 1 by a non-trivial factor.

4.1 Implications in Fair Division

Despite the fact that the Round-Robin allocation protocol is known to produce EF1 allocations for agents with additive valuation functions and no constraints, showing analogous guarantees in our setting is non-trivial. Theorems 4.4 and 4.5 are in this direction and have implications on constrained fair division as we will see below. Note that the analysis of Theorem 4.5 is tight, as it is known that Round-Robin as an algorithm cannot guarantee more than 0.5-EF1 for submodular valuation functions even without constraints [2].

Theorem 4.4. *Let i be an agent with a p_i-system constraint $\mathcal{I}_i$ who chooses greedily in the Round-Robin protocol. Also, let j be any other agent and g be the first item added to S_j. Then,*

$$f_i(S_i) \geq \frac{1}{p_i+2} \cdot \max_{S \in \mathcal{I}_i | S'_j} f_i(S),$$

where $S'_j = S_j$ if $i < j$, and $S'_j = S_j \setminus \{g\}$ otherwise.

Theorem 4.5. *Let i be an agent with a cardinality constraint $\mathcal{I}_i$ who chooses greedily in the Round-Robin protocol. Also, let j be any other agent and g be the first item added to S_j. Then,*

$$f_i(S_i) \geq 0.5 \cdot \max_{S \in \mathcal{I}_i | S'_j} f_i(S)$$

where $S'_j = S_j$ if $i < j$, and $S'_j = S_j \setminus \{g\}$ otherwise.

If the items in M were allocated in a centralized way, one could interpret the monotone case of our problem as a fair division problem where the agents have submodular valuation functions as well as feasibility constraints over the subsets of goods. The problem of fairly allocating goods subject to cardinality, matroid, or even more general, constraints has been studied before; see Suksompong [34] for a recent survey. Also, there is a rich line of work on discrete fair division beyond additive valuation functions, e.g., [2, 11, 17, 23, 24]; see also [1] and references therein.

Here we focus exclusively on (an appropriate generalization of) *envy-freeness up to an item* (EF1), a notion which was introduced by Budish [15] (and, implicitly, by Lipton et al. [29] a few years earlier). An *allocation* is a tuple of disjoint subsets of M, $A = (A_1, \ldots, A_n)$, such that each agent $i \in [n]$ receives the set A_i. Note that here we do not assume the allocation to be complete, i.e., $\bigcup_{i\in[n]} A_i = M$, because of the presence of the feasibility constraints in our setting.

Definition 4.6. An allocation A is *α-approximate envy-free up to one item (EF1)* if, for every pair of agents $i, j \in [n]$, there is some $g \in A_j$, such that $v_i(A_i) \geq \alpha\, v_i(A_j \setminus \{g\})$.

When $\alpha = 1$, we just refer to EF1 allocations. Round-Robin, if implemented as an algorithm with all agents choosing greedily, is known to produce EF1 allocations when agents have additive valuation functions [16] and 0.5-EF1 allocations when agents have submodular valuation functions [2], assuming no constraints at all. Although it is possible to have EF1 allocations under cardinality constraints for agents with additive valuation functions [12], this definition that ignores feasibility is way too strong for our general constraints. What we need here is the notion of *feasible EF1* introduced recently by Dror et al. [19] and others (see, e.g., Barman et al. [10]), or rather its approximate version.

Definition 4.7. An allocation A is *α-approximate feasible envy-free up to one item (α-FEF1)* if, for every pair of agents $i, j \in [n]$, there is some $g \in A_j$, such that $v_i(A_i) \geq \alpha\, v_i(A'_j)$ for any $A'_j \subseteq A_j \setminus \{g\}$ that is feasible for agent i.

By simulating a run of Round-Robin($\boldsymbol{\mathcal{G}}$), i.e., our Protocol 1 with greedy policies for all agents, Theorems 4.4 and 4.5 give us the following direct corollaries.

Corollary 4.8. *For agents with submodular valuation functions and p-system constraints, we can efficiently find a $\frac{1}{p+2}$-FEF1 allocation, such that adding any unallocated item to any agent's set is infeasible.*

Corollary 4.9. *For agents with submodular valuation functions and cardinality constraints, we can efficiently find a 0.5-FEF1 allocation, such that the allocation is complete or the maximum cardinalities of all agents have been met.*

Similarly to Theorem 4.5, Corollary 4.9 is tight even without constraints [2]. We suspect that the tight result for Corollary 4.8 would be $p_i + 1$ instead.

4.2 Improved Guarantees for Robust Instances

A natural next question now is: *is it possible for an agent to choose items optimally—or at least, in a way that significantly improves over choosing greedily?* In Section 5, we show that this poses a computationally hard task. Before doing so, however, we divert the question to *under what circumstances* is it possible for an agent to choose items (approximately) optimally by choosing greedily. We turn our attention to the special case of (α, β)-robust instances (see Definition 2.4), for which the results of Theorems 4.1 and 4.3 can be significantly improved.

We begin with a parameterized result about a specific agent, which will imply, as a corollary, a strong guarantee for instances that are $(\Omega(n), O(1))$-robust with respect to everyone. Note that γ and β in the following statement can be functions of n; in particular, we assume that γ is such that $\gamma n + i - 1$ is always an integer.

Theorem 4.10. *Assume that an instance is $(\gamma n + i - 1, \beta)$-robust with respect to agent i (who has a p_i-system constraint). If i chooses greedily in the Round-Robin protocol, they build a solution S_i such that $f_i(S_i) \geq \mathrm{OPT}_i / \beta(p_i + 1 + 1/\gamma)$.*

Proof. Our notation will be consistent with the one introduced in the proof of Theorem 4.1. That is, item x^ℓ_j is the j-th item that agent ℓ chooses from the moment when agent i is about to start choosing, $S^{(r)}_i$ is the solution of agent i right before item x^i_r is added to it, and $s := |S_i|$.

Since the instance is $(\gamma n + i - 1, \beta)$-robust with respect to i, by the time i gets to pick their first item, there are at least $a := \gamma n$ disjoint subsets of M, say $Q_{i1}, \ldots, Q_{ia}$, such that $Q_{ij} \in \mathcal{I}_i$ and $\beta f_i(Q_{ij}) \geq \mathrm{OPT}_i$, for any $j \in [a]$, *and* no item in $\cup_{j=1}^a Q_{ij}$ has been allocated to any of the agents 1 through $i - 1$ yet.

Here we invoke the full power of Lemma 4.2, as we apply it to all $Q_{i1}, \ldots, Q_{ia}$ and assume the existence of δ as in the statement of the lemma. Further, we apply the second part of Theorem 2.5 for S_i and each Q_{ij}, and we add up these a inequalities:

$$\begin{aligned}
\sum_{j=1}^{a} f_i(Q_{ij}) &\leq \sum_{j=1}^{a} f_i(S_i) + \sum_{j=1}^{a} \sum_{x \in Q_{ij} \setminus S_i} f_i(x \mid S_i) \\
&\leq a f_i(S_i) + \sum_{r=1}^{s} \sum_{x \in \delta^{-1}(x^i_r)} f_i(x \mid S_i) \\
&\leq a f_i(S_i) + \sum_{r=1}^{s} \sum_{x \in \delta^{-1}(x^i_r)} f_i(x \mid S^{(r)}_i)
\end{aligned}$$

$$
\begin{aligned}
&\le a f_i(S_i) + \sum_{r=1}^{s} (n + a p_i - 1) f_i(x_r^i \,|\, S_i^{(r)}) \\
&\le \gamma n f_i(S_i) + (n + \gamma n p_i) f_i(S_i) = n(\gamma + 1 + \gamma p_i) f_i(S_i)\,,
\end{aligned}
$$

where the second inequality holds because we have $\bigcup_{r=1}^{s} \delta^{-1}(x_r^i) = \left(\bigcup_{j\in[a]} Q_{ij}\right) \setminus S_i$, the third follows from submodularity, and the fourth from the second property of Lemma 4.2.

On the other hand, by the definition of Q_{ij}s, we have

$$
\sum_{j=1}^{a} f_i(Q_{ij}) \ge \sum_{j=1}^{a} \frac{\mathrm{OPT}_i}{\beta} = \frac{\gamma n}{\beta}\mathrm{OPT}_i\,.
$$

We conclude that $f_i(S_i) \ge \mathrm{OPT}_i/\beta(p_i + 1 + 1/\gamma)$, as claimed. □

Although Theorem 4.10 is agent specific, one would expect that, in applications like our influence maximization running example, typical real-world instances are extremely robust with respect to everyone. The rationale here is that the set M contains much more value than what an agent can extract given their constraint. The following corollary gives strong guarantees with respect to all agents for such scenarios. Recall that these guarantees are almost the best one could hope for in polynomial time [9].

Corollary 4.11. *Assume that an instance is* $(\lceil(1+\gamma)n\rceil, \beta)$*-robust with respect to every agent for constant* $\beta, \gamma \in \mathbb{R}_+$*. Any agent* i *with a* p_i*-system constraint who chooses greedily in the Round-Robin protocol, achieves a* $1/\Theta(p_i)$ *fraction of* OPT_i*.*

A final remark here is that our definition of (α, β)-robustness and Theorem 4.10 could possibly lead to simple protocols with approximate *maximin share* guarantees. In particular, we suspect that a variant of Algorithm 3 of Barman and Krishnamurthy [11] (which essentially falls under our notion of a protocol), combined with a modified version of our Theorem 4.10, should imply $1/\Theta(p_i)$-approximate guarantees with respect to each agent's maximin share.

5 IT IS HARD TO OPTIMIZE OVER OTHERS

So far, there was no particular need to talk about computational efficiency. The greedy policy clearly runs in polynomial time, assuming polynomial-time value and independence oracles, and the Round-Robin protocol itself delegates any non-trivial computational task to the agents themselves. Here we need to clarify our terminology a bit. When we refer to polynomial-time algorithms in the statements of the next two theorems we mean algorithms that determine the choices of a single agent, who has full information about all the objective functions, and whose number of steps and number of queries to submodular function value oracles are bounded by a polynomial in the number of agents. Within their proofs, however, we reduce the existence of such algorithms to polynomial-time approximation algorithms for constrained submodular maximization, whose number of steps and number of queries to a submodular function value oracle are bounded by a polynomial in the size of the ground set over which the submodular function is defined.

When we say that an algorithm $\mathcal{A}_i$ is a ρ*-improvement of agent* i *over the greedy policy* $\mathcal{G}_i$ *(given the objective functions and policies of all other agents)*, we mean that running $\mathcal{A}_i$, while the other agents stick to their corresponding policies, results in agent i receiving a set ρ times more valuable than the one they receive by choosing greedily, *whenever this is possible.*

The main result here is Theorem 5.1. On a high level, we show that in this setting the inapproximability result of Feige [20] is amplified to the point where consistently doing *slightly* better than the greedy policy is NP-hard, even when other agents pick elements in the most predictable way. Specifically, this hardness result holds even when there are no individual feasibility constraints and when all other agents (i.e., everyone except the agent to whom the hardness applies to) use strategies as simple as choosing greedily. That is, the hardness does not stem from having strong constraints or hard to analyze policies for everyone else, but rather indicates the inherent computational challenges of the problem. We find this to be rather counter-intuitive. The proof of Theorem 5.1 is not straightforward either. To prove the theorem we do not construct a general reduction as usual, but rather we deal with a number of cases algorithmically and we only use a reduction for instances with very special structure.

Theorem 5.1. *Assume* $n \ge 2$*, let* $\varepsilon \in (0, 0.3)$ *be any small constant, and fix any* $j \in [n]$*. Even in instances where all agents in* $[n] \setminus \{j\}$ *have additive objective functions and greedy policies, there is no polynomial-time algorithm that is a* $(1+\varepsilon)$*-improvement of agent* j *over the greedy policy* $\mathcal{G}_j$*, unless* P = NP.

6 DEALING WITH NON-MONOTONICITY

When trying to maximize a non-monotone submodular function—with or without constraints—a naively greedy solution may be arbitrarily away from a good solution even in the standard setting with a single agent [4]. Several approaches have been developed to deal with non-monotonicity, but the recently introduced *simultaneous greedy* approach [5, 21] seems like a natural choice here. The high-level idea is to bypass the complications of non-monotonicity by simultaneously constructing multiple greedy solutions. Here we consider building only *two* greedy solutions, not just for the sake of simplicity but also because the technical analysis for multiple solutions does not translate well into our setting. Although we suspect that a more elaborate approach with more solutions could shave off a constant factor asymptotically, we see the results of this section as a proof of concept that a simple greedy policy can work well in the non-monotone version of our problem as well.

Policy 2 Simultaneous Greedy policy $\mathcal{G}_i^+(S_{i1}, S_{i2}\,; Q)$ of agent i.
(S_{i1}, S_{i2} : current solutions of i (initially $S_{i1} = S_{i2} = \emptyset$) ; Q : current set of available items)

```
A = {(x, y) ∈ Q × {1, 2} : S_iy ∪ {x} ∈ I_i}
if A ≠ ∅ then
    Let (j, ℓ) ∈ arg max_{(z,w)∈A} f(z | S_iw)
    S_iℓ = S_iℓ ∪ {j}
    return j
else
    return a dummy item                (i.e., return nothing)
```

Coming back to building two greedy solutions, an agent i maintains two sets S_{i1}, S_{i2} and every time that it is their turn to pick an item, they pick a single item that maximizes the marginal value with respect to either S_{i1} or S_{i2} among the items that are still available and for which adding them to the respective solution is feasible. Formally, saying that agent i chooses greedily, now means that they choose according to the policy $\mathcal{G}_i^+$ given above. As we did in the monotone case, for cardinality constraints we can get a somewhat stronger guarantee.

Theorem 6.1. *Any agent i with a non-monotone objective and a p_i-system (resp. cardinality) constraint, who follows the greedy policy $\mathcal{G}_i^+$ in Protocol 1, builds solutions S_{i1}, S_{i2} such that* $\max_{t\in\{1,2\}} f_i(S_{it}) \geq \mathrm{OPT}_i^- / (4n + 4p_i + 2)$ *(resp.* $\max_{t\in\{1,2\}} f_i(S_{it}) \geq \mathrm{OPT}_i^- / (4n + 2)$*).*

6.1 Improved Guarantees for Robust Instances

Like in Section 4.2 for the monotone case, here we explore what is possible for robust instances. We show that the linear factors of Theorem 6.1 can still be removed for instances that are $(\Omega(n), O(1))$-robust with respect to everyone. We state the analogs of Theorem 4.10 and Corollary 6.3.

Theorem 6.2. *Assume an instance is $(\gamma n + i - 1, \beta)$-robust with respect to agent i (having a non-monotone objective and a p_i-system constraint). By choosing greedily in the Round-Robin protocol, i builds S_{i1}, S_{i2} such that* $\max_{t\in\{1,2\}} f_i(S_{it}) \geq \mathrm{OPT}_i / 2\beta(2p_i + 1 + 2/\gamma)$*.*

Corollary 6.3. *If an instance is $(\lceil (1+\gamma)n \rceil, \beta)$-robust with respect to every agent for constant $\beta, \gamma \in \mathbb{R}_+$, then any agent i with a non-monotone objective and a p_i-system constraint who chooses greedily in the Round-Robin protocol, achieves a $1/\Theta(p_i)$ fraction of* OPT_i*.*

7 IMPROVING ALGORITHMIC FAIRNESS AND GUARANTEES VIA RANDOMNESS

In this section we rectify the obvious shortcoming of Protocol 1, namely that not all agents are treated equally due to their fixed order. Unfortunately, this inequality issue is inherent to any deterministic protocol which is agnostic to the objective functions (as Protocol 1 is) and it heavily affects agents in the presence of a small number of highly valued contested items. A natural remedy, which we apply here, is to randomize over the initial ordering of the agents before running the main part of Protocol 1. This *Randomized Round-Robin* protocol is formally described in Protocol 2:

Protocol 2 Randomized Round-Robin($\mathcal{A}_1, \ldots, \mathcal{A}_n$)
(For $i \in [n]$, $\mathcal{A}_i$ is the policy of agent i.)

Let $\pi : [n] \to [n]$ be a random permutation on $[n]$
$Q = M$; $k = \lceil m/n \rceil$
for $r = 1, \ldots, k$ **do**
  for $i = 1, \ldots, n$ **do**
    $j = \mathcal{A}_{\pi(i)}(S_{\pi(i)}\, ; Q)$ (j could be a *dummy* item)
    $Q = Q \setminus \{j\}$

Of course, given a permutation π, all the guarantees of Theorems 4.1, 4.3, and 6.1 still hold ex-post, albeit properly restated. That is, now the guarantee for an agent $i \in N$ are with respect to the value of an optimal solution available to i, given that $\pi^{-1}(i) - 1$ items have been lost to agents $\pi(1), \pi(2), \ldots, \pi(\pi^{-1}(i) - 1)$ before i gets to pick their first item.

The next theorem states that, in every worst-case scenario we studied in Sections 4 and 6, all agents who choose greedily obtain a set of expected value within a constant factor of the best possible worst-case guarantee of OPT_i/n (recall the example from the Introduction about this being best possible).

Theorem 7.1. *Assume agent i chooses greedily in the Randomized Round-Robin protocol. Then, the expected value that i obtains (from the best solution they build) is at least* $\mathrm{OPT}_i/\beta\, n$*, where $\beta = 2 + p_i/n$ (resp. $\beta = 5 + (4p_i + 2)/n$), if i has a p_i-system constraint and a monotone (resp. non-monotone) submodular objective.*

In the second result of this section, we show that the assumptions needed (in terms of robustness) to obtain $O(p_i)$ approximation guarantees with respect to OPT_i, are significantly weaker. In particular, Theorem 7.2 provides asymptotically best-possible guarantees for *any* instances that are $(\Omega(n), O(1))$-robust with respect to everyone, no matter what the hidden constants are.

Theorem 7.2. *Assume that an instance is $(\lceil \delta n \rceil, \beta)$-robust with respect to every agent for constant $\beta, \delta \in \mathbb{R}_+$. Also, assume that any agent i, who has a p_i-system constraint and a submodular objective, chooses greedily in the Randomized Round-Robin protocol. Then, in expectation, i achieves a $1/\Theta(p_i)$ fraction of* OPT_i*.*

8 DISCUSSION AND OPEN QUESTIONS

In this work we studied a combinatorial optimization problem, where the goal is the non-centralized constrained maximization of multiple submodular objective functions. We showed that Round-Robin as a coordinated maximization protocol exhibits very desirable properties, in terms of individual approximation guarantees, ease of participation (greedy is straightforward and, in some sense, the best way to go), as well as transparency (simple, non-adaptive protocol).

The most natural direction for future work is the design and study of different protocols that perform better or have different benchmarks, e.g., a *maximin share* type of benchmark. Ideally, we would like such protocols to be easily implemented and at the same time to provide strong approximation guarantees for agents that follow easy and simple policies. Also, recall that opposed to traditional settings, we assumed no knowledge about the functions or constraints. It would be interesting to also explore adaptive protocols that perform part of the computational work centrally. A closely related question is whether the improved approximations of such protocols, could imply stronger fairness guarantees with respect to EF1 or other fairness notions. Finally, regardless of the protocol, another meaningful direction would be to explore what can be achieved in terms of approximation for objective functions in richer classes (e.g., subadditive).

ACKNOWLEDGMENTS

This work was supported by the ERC Advanced Grant 788893 AMDROMA “Algorithmic and Mechanism Design Research in Online Markets”, the MIUR PRIN project ALGADIMAR “Algorithms, Games, and Digital Markets”, the NWO Veni project VI.Veni.192.153, and the project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

REFERENCES

[1] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A. Voudouris, and Xiaowei Wu. 2023. Fair division of indivisible goods: Recent progress and open questions. *Artif. Intell.* 322 (2023), 103965.

[2] Georgios Amanatidis, Georgios Birmpas, Philip Lazos, Stefano Leonardi, and Rebecca Reiffenhäuser. 2023. Round-Robin Beyond Additive Agents: Existence and Fairness of Approximate Equilibria. In *Proceedings of the 24th ACM Conference on Economics and Computation, EC 2023*. ACM, 67–87.

[3] Georgios Amanatidis, Georgios Birmpas, Philip Lazos, Stefano Leonardi, and Rebecca Reiffenhäuser. 2024. Algorithmically Fair Maximization of Multiple Submodular Objective Functions. *CoRR* abs/2402.15155 (2024). https://doi.org/10.48550/arXiv.2402.15155

[4] Georgios Amanatidis, Federico Fusco, Philip Lazos, Stefano Leonardi, and Rebecca Reiffenhäuser. 2022. Fast Adaptive Non-Monotone Submodular Maximization Subject to a Knapsack Constraint. *J. Artif. Intell. Res.* 74 (2022), 661–690.

[5] Georgios Amanatidis, Pieter Kleer, and Guido Schäfer. 2022. Budget-Feasible Mechanism Design for Non-monotone Submodular Objectives: Offline and Online. *Math. Oper. Res.* 47, 3 (2022), 2286–2309.

[6] Georgios Amanatidis, Evangelos Markakis, and Apostolos Ntokos. 2020. Multiple birds with one stone: Beating 1/2 for EFX and GMMS via envy cycle elimination. *Theor. Comput. Sci.* 841 (2020), 94–109.

[7] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. 2022. Fair allocation of indivisible goods and chores. *Autonomous Agents and Multi Agent Systems* 36, 1 (2022), 3.

[8] Haris Aziz, Paul Goldberg, and Toby Walsh. 2017. Equilibria in Sequential Allocation. In *Proceedings of the 5th International Conference on Algorithmic Decision Theory, ADT '17 (LNCS, Vol. 10576)*. Springer, 270–283.

[9] Ashwinkumar Badanidiyuru and Jan Vondrák. 2014. Fast algorithms for maximizing submodular functions. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*. SIAM, 1497–1514.

[10] Siddharth Barman, Arindam Khan, Sudarshan Shyam, and K. V. N. Sreenivas. 2023. Finding Fair Allocations under Budget Constraints. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023*. AAAI Press, 5481–5489.

[11] Siddharth Barman and Sanath Kumar Krishnamurthy. 2020. Approximation Algorithms for Maximin Fair Division. *ACM Trans. Economics and Comput.* 8, 1 (2020), 5:1–5:28.

[12] Arpita Biswas and Siddharth Barman. 2018. Fair Division Under Cardinality Constraints. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, Jérôme Lang (Ed.). ijcai.org, 91–97.

[13] Adam Breuer, Eric Balkanski, and Yaron Singer. 2020. The FAST algorithm for submodular maximization. In *International Conference on Machine Learning*. PMLR, 1134–1143.

[14] Niv Buchbinder and Moran Feldman. 2018. Submodular Functions Maximization Problems. In *Handbook of Approximation Algorithms and Metaheuristics, Second Edition, Volume 1: Methologies and Traditional Applications*. Chapman and Hall/CRC, 753–788.

[15] Erik Budish. 2011. The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes. *Journal of Political Economy* 119, 6 (2011), 1061–1103.

[16] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. 2019. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Trans. Economics and Comput.* 7, 3 (2019), 12:1–12:32.

[17] Chandra Chekuri, Pooja Kulkarni, Rucha Kulkarni, and Ruta Mehta. 2024. 1/2-Approximate MMS Allocation for Separable Piecewise Linear Concave Valuations. In *38th AAAI Conference on Artificial Intelligence, AAAI 2024*. AAAI Press, 9590–9597.

[18] Yixin Chen and Alan Kuhnle. 2023. Approximation Algorithms for Size-Constrained Non-Monotone Submodular Maximization in Deterministic Linear Time. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023*. ACM, 250–261.

[19] Amitay Dror, Michal Feldman, and Erel Segal-Halevi. 2023. On Fair Division under Heterogeneous Matroid Constraints. *J. Artif. Intell. Res.* 76 (2023), 567–611.

[20] Uriel Feige. 1998. A Threshold of ln n for Approximating Set Cover. *J. ACM* 45, 4 (1998), 634–652.

[21] Moran Feldman, Christopher Harshaw, and Amin Karbasi. 2023. How Do You Want Your Greedy: Simultaneous or Repeated? *J. Mach. Learn. Res.* 24 (2023), 72:1–72:87.

[22] Marshall L. Fisher, George L. Nemhauser, and Laurence A. Wolsey. 1978. An analysis of approximations for maximizing submodular set functions - II. *Mathematical Programming Study* 8 (1978), 73–87.

[23] Jugal Garg, Pooja Kulkarni, and Rucha Kulkarni. 2023. Approximating Nash Social Welfare under Submodular Valuations through (Un)Matchings. *ACM Trans. Algorithms* 19, 4 (2023), 36:1–36:25.

[24] Mohammad Ghodsi, Mohammad Taghi Hajiaghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. 2022. Fair allocation of indivisible goods: Beyond additive valuations. *Artif. Intell.* 303 (2022), 103633.

[25] Anupam Gupta, Aaron Roth, Grant Schoenebeck, and Kunal Talwar. 2010. Constrained Non-monotone Submodular Maximization: Offline and Secretary Algorithms. In *Internet and Network Economics - 6th International Workshop, WINE 2010. Proceedings (LNCS, Vol. 6484)*. Springer, 246–257.

[26] Kai Han, Zongmai Cao, Shuang Cui, and Benwei Wu. 2020. Deterministic Approximation for Submodular Maximization over a Matroid in Nearly Linear Time. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.

[27] Alan Kuhnle. 2019. Interlaced Greedy Algorithm for Maximization of Submodular Functions in Nearly Linear Time. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*. 2371–2381.

[28] David Kurokawa. 2017. *Fair Division in Game Theoretic Settings*. Ph.D. Dissertation. Carnegie Mellon University.

[29] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. 2004. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*. 125–131.

[30] Pasin Manurangsi and Warut Suksompong. 2021. Closing Gaps in Asymptotic Fair Division. *SIAM Journal on Discrete Mathematics* 35, 2 (2021), 668–706.

[31] Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, and Amin Karbasi. 2016. Fast Constrained Submodular Maximization: Personalized Data Summarization. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016 (JMLR, Vol. 48)*. JMLR.org, 1358–1367.

[32] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.* 14, 1 (1978), 265–294.

[33] Canh V. Pham, Tan D. Tran, Dung T. K. Ha, and My T. Thai. 2023. Linear Query Approximation Algorithms for Non-monotone Submodular Maximization under Knapsack Constraint. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023*. ijcai.org, 4127–4135.

[34] Warut Suksompong. 2021. Constraints in fair division. *SIGecom Exch.* 19, 2 (2021), 46–61.

[35] Xiaoming Sun, Jialin Zhang, Shuo Zhang, and Zhijie Zhang. 2022. Improved Deterministic Algorithms for Non-monotone Submodular Maximization. In *Computing and Combinatorics - 28th International Conference, COCOON 2022, Proceedings (LNCS, Vol. 13595)*. Springer, 496–507.

[36] Biaoshuai Tao and Mingwei Yang. 2024. Fair and Almost Truthful Mechanisms for Additive Valuations and Beyond. *CoRR* abs/2306.15920 (2024). https://doi.org/10.48550/arXiv.2306.15920