Cost Sharing over Combinatorial Domains: Complement-Free Cost Functions and Beyond

Georgios Birmpas
Department of Computer Science, University of Oxford, UK
Department of Informatics, Athens University of Economics and Business, Greece
gebirbas@gmail.com

Evangelos Markakis
Department of Informatics, Athens University of Economics and Business, Greece
markakis@aueb.gr

Guido Schäfer
Networks and Optimization group, Centrum Wiskunde & Informatica (CWI), The Netherlands
Dept. of Econometrics and Operations Research, Vrije Universiteit Amsterdam, The Netherlands
g.schaefer@cwi.nl

──── **Abstract** ────

We study mechanism design for combinatorial cost sharing models. Imagine that multiple items or services are available to be shared among a set of interested agents. The outcome of a mechanism in this setting consists of an assignment, determining for each item the set of players who are granted service, together with respective payments. Although there are several works studying specialized versions of such problems, there has been almost no progress for general combinatorial cost sharing domains until recently [7]. Still, many questions about the interplay between strategyproofness, cost recovery and economic efficiency remain unanswered.

The main goal of our work is to further understand this interplay in terms of budget balance and social cost approximation. Towards this, we provide a refinement of cross-monotonicity (which we term *trace-monotonicity*) that is applicable to iterative mechanisms. The trace here refers to the order in which players become finalized. On top of this, we also provide two parameterizations (complementary to a certain extent) of cost functions which capture the behavior of their average cost-shares. Based on our trace-monotonicity property, we design a scheme of ascending cost sharing mechanisms which is applicable to the combinatorial cost sharing setting with symmetric submodular valuations. Using our first cost function parameterization, we identify conditions under which our mechanism is weakly group-strategyproof, $O(1)$-budget-balanced and $O(H_n)$-approximate with respect to the social cost. Further, we show that our mechanism is budget-balanced and H_n-approximate if both the valuations and the cost functions are symmetric submodular; given existing impossibility results, this is best possible. Finally, we consider general valuation functions and exploit our second parameterization to derive a more fine-grained analysis of the Sequential Mechanism introduced by Moulin. This mechanism is budget balanced by construction, but in general only guarantees a poor social cost approximation of n. We identify conditions under which the mechanism achieves improved social cost approximation guarantees. In particular, we derive improved mechanisms for fundamental cost sharing problems, including Vertex Cover and Set Cover.

1   Introduction

How to share the cost of a common service (or public good) among a set of interested agents constitutes a fundamental problem in mechanism design that has been studied intensively for at least two decades. Several deep and significant advancements have been achieved throughout this period, notably also combining classical mechanism design objectives (such as incentive compatibility, economic efficiency, etc.) with theoretical computer science objectives (such as approximability and computational efficiency).

However, in the vast majority of the cost sharing models that have been proposed and analyzed in the literature, it is assumed that the mechanism designer is offering a single service and that each agent has a private value describing the willingness to pay for the service. At the same time, there is also a publicly known cost function which describes the total cost for offering the service to each possible subset of agents. Said differently, this results in a single-parameter mechanism design problem, where the goal is to select a subset of the players that will be granted service, subject to covering the cost and achieving an economically efficient outcome.

Although significant progress has been made for such single-parameter domains, moving towards more general *combinatorial domains* has been almost elusive so far. Imagine that there are multiple goods to be shared among the agents who now have more complex valuation functions, expressing their willingness to pay for different subsets (or bundles) of goods. The cost function now depends on the subsets of agents sharing each of the items. An outcome of a mechanism under this setting, consists of an allocation, which specifies for each agent the goods for which she is granted service, together with a payment scheme.

The desired properties in designing a cost-sharing mechanism (be it combinatorial or not) are three-fold: (i) *group-strategyproofness*: we would like resistance to misreporting preferences by individual agents or coalitions, (ii) *budget-balance*: the payments of the players should cover the incurred cost, (iii) *economic efficiency*: the allocation should maximize a measure of social efficiency. The fundamental results in [10, 16] rule out the possibility that all three properties can be achieved. As a result, if we insist on any variant of strategyproofness, we are forced to settle with approximate notions of at least one of the other two criteria. In this context, approximate budget balance means that the mechanism may overcharge the agents, but not by too much. In terms of efficiency, considering a social cost objective instead of the classical social welfare objective (definitions are given in Section 2) seems more amenable for multiplicative approximation guarantees.

These adapted objectives have been investigated thoroughly for single-parameter problems, especially for cost-sharing variants of well-known optimization problems. In the context of more general combinatorial cost-sharing mechanisms, a restricted model with multiple levels of service was first studied in [13]. Ever since, for almost a decade, there was no additional progress along these lines. It was only recently that a step forward was made by Dobzinski and Ovadia [7]. In their work, they introduce a combinatorial cost-sharing model and derive the first mechanisms guaranteeing good budget balance and social cost approximation guarantees for different classes of valuation and cost functions. As already pointed out in [7], however, several important questions concerning our understanding of the approximability of these objectives remain open and deserve further study. This constitutes the starting point of our investigations reported in this work.

Our Contributions.   We make further advancements on the design and analysis of mechanisms for combinatorial cost-sharing models. To begin with, the analysis of the mechanisms we study asks for new conceptual ideas which might be interesting on their own:

- We first provide a refinement of the well-known notion of *cross-monotonic* cost sharing functions, which is key in the intensively studied class of Moulin-Shenker mechanisms [15] for the single-parameter domain. We introduce the notion of *trace-monotonic* cost sharing functions which is applicable for mechanisms that proceed iteratively and evict agents one-by-one. Trace-monotonicity formalizes the fact that the cost-shares observed by a player for an item do not decrease throughout the *course* of the mechanism. That is, these cost shares may depend on the specific order (or *trace* as we will call it) in which the mechanism considers the agents.

- We identify two different and (to some extent) complementary parameterizations of the cost functions. Intuitively, these parameters measure the "variance" of the average cost-share $c(S)/|S|$, over all agent sets S. We introduce two such notions, which we term *α-average decreasing* and *α-average min-bounded* (see Definition 5 and Definition 15, respectively). We note that for every cost function, there exist respective values of α (possibly different for each definition) for which these properties are satisfied. These definitions provide an alternative way to classify cost functions and their respective approximation guarantees in terms of budget balance and social cost.

Using the above ideas, in Section 3, we derive a scheme for ascending cost sharing mechanisms, which can be seen as a (non-trivial) adaptation of the Moulin-Shenker mechanisms from the binary accept/reject setting to combinatorial cost sharing. Our notion of trace-monotonic cost shares plays a crucial role here. We show that our proposed mechanism is applicable for any non-decreasing cost function and for symmetric submodular valuations (i.e., submodular functions whose value depends only on the cardinality of the set).

By exploiting the first parameterization of α-average decreasing cost functions, our main result of Section 3 is that for $\alpha = O(1)$, our mechanism is polynomial-time, weakly group-strategyproof, $O(1)$-budget-balanced and $O(H_n)$-approximate with respect to social cost, where n is the number of agents.[1] As a consequence, if both the valuation and cost functions are symmetric submodular ($\alpha = 1$), the mechanism is budget-balanced and H_n-approximate. This is best possible even for a single item, as there exist corresponding inapproximability results by Dobzinski et al. [6]. Prior to our work, the best known mechanism for symmetric submodular valuation and cost functions is H_n-budget balanced and H_n-approximate [7]. We anticipate that further extensions and generalizations might be feasible through our framework and this type of ascending mechanisms.

In Section 4, we exploit our second parameterization of α-average min-bounded cost functions, and provide results for general valuation functions. As it turns out, our parameterization enables us to obtain a more fine-grained analysis of the Sequential Mechanism introduced by Moulin [14]. This mechanism is budget-balanced by construction, but in general only guarantees a poor social cost approximation of factor n. We show that for α-average min-bounded cost functions with $\alpha = O(1)$, the Sequential Mechanism is budget balanced and H_n-approximate with respect to social cost. Interestingly, this result does not even require monotonicity of the valuation functions. In addition, we can push our results even a bit further by introducing a refinement of this class of cost functions (see Definition 20) for which we show that the Sequential Mechanism is $O(1)$-approximate. The refinement allows us to obtain improved mechanisms for several cost functions originating from combinatorial optimization problems. For example, our result implies that the Sequential Mechanism is d-approximate for certain cost-sharing variants of Vertex Cover and Set Cover, where d is the maximum degree of a node or the maximum size of a set, respectively; this improves upon existing results, even in the well-studied single-item case, when d is constant.

---

[1] We use H_n to denote the *n-th Harmonic number* defined as $H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n}$.

In general, the two parameterizations of the cost functions introduced in this work seem to be suitable means to accurately capture the approximation guarantees of both the ascending cost-sharing mechanism of Section 3, and the Sequential Mechanism of Section 4. In fact, we have not managed to construct natural examples of cost functions which do not admit an $O(H_n)$-approximation by neither of the mechanisms studied here. See also the discussion in Section 5, where some examples are constructed but they are rather artificial (Proposition 25). As such, these parameterizations help us to narrow down the class of cost functions which are not yet known to admit a good social cost approximation and enhance our understanding towards further progress in combinatorial cost sharing.

Related Work.   For the single-item setting and with submodular cost functions, the best known group-strategyproof and budget balanced cost-sharing mechanism is arguably the Shapley value mechanism, introduced by Moulin and Shenker [14, 15]. This was also the first work that tried to quantify the efficiency loss of budget balanced cost-sharing mechanisms. Later, Feigenbaum et al. [9] showed that if one insists on truthfulness, there is no mechanism that achieves a finite approximation of the social welfare objective, even if one relaxes the budget balance property to cost recovery. To overcome this impossibility result, Roughgarden et al. [17] introduced the notion of *social cost* as an alternative means to quantify the efficiency of a mechanism. In the same work, they showed that the Shapley value mechanism is H_n-approximate with respect to this objective. Dobzinski et al. [6] established another impossibility result for the social cost objective, and showed that every mechanism satisfying truthfulness and cost recovery cannot achieve a social cost approximation guarantee better than $\Omega(\log n)$. The problem of deriving mechanisms with the best possible budget balance and social cost approximation guarantees for different cost functions arising from combinatorial optimization problems has been extensively studied in various works, see e.g., [2, 3, 4, 11].

Moving beyond the single-item case, Mehta et al. [13] introduced a new family of truthful mechanisms (called *acyclic mechanisms*) which apply to general demand settings of multiple identical items when players have symmetric submodular valuations. For additional works that consider the general demand setting, the reader is referred to [2, 3, 5, 14]. Birmpas et al. [1] also studied families of valuation and cost functions for the multiple item setting, under cost sharing models that are motivated by applications in participatory sensing environments.

Most related to our work is the recent work by Dobzinski and Ovadia [7]. To the best of our knowledge, this is the only prior work that considers a more general approach for combinatorial cost sharing. They studied a multi-parameter setting and proposed a new VCG-based mechanism. Basically, the idea is to run a VCG mechanism with respect to a modified objective function which is defined as the sum of the player valuations minus a potential. Intuitively, the latter ensures that the payments computed by the mechanism cover the actual cost. They showed that this mechanism is strategyproof and H_n-approximate with respect to social cost. They also identified several classes of valuation and cost functions for which the mechanism is H_n-budget balanced. In particular, this is the case if the valuation and cost functions are symmetric.[2] Additionally, they established that their mechanism is optimal with respect to the social cost approximation among all symmetric VCG-based mechanisms that always cover the cost.

----

[2] We note that their definition of symmetry for the cost function differs from the one we use here.

2 Definitions and Notation

We assume there is a set $N = \{1, 2, \ldots, n\}$ of players and a set $M = \{1, 2, \ldots, m\}$ of items. Each item can be viewed as a public good or some service that can be shared by the players. Each player i has a *private valuation function* $v_i : 2^M \to \mathbb{R}_{\geq 0}$ specifying the value that she derives from each subset of items.

A *cost-sharing mechanism* takes as input the declared (possibly false) valuation functions $\vec{b} = (b_i)_{i \in N}$ of the players and outputs (i) an allocation that determines which players share each item and (ii) a payment p_i for each player i. An allocation is denoted by a tuple $A = (A_1, \ldots, A_n)$, where $A_i \subseteq M$ is the set of items provided to player i. For notational convenience, we also represent an allocation $A = (A_i)_{i \in N}$ as a tuple over the items space $(T_1, \ldots, T_m)$ such that for every item $j \in M$, $T_j \subseteq N$ is the subset of players sharing item j, i.e., $T_j = \{i \in N : j \in A_i\}$.

In this paper, we consider mostly *separable* cost functions. In the *separable setting*, we assume that the overall cost of an allocation decomposes into the cost for providing each item separately. Hence, every item j is associated to a known cost function $c_j : 2^N \to \mathbb{R}_{\geq 0}$, which specifies for each set of players $T \subseteq N$, the cost $c_j(T)$ of providing item j to the players in T. Thus, the total cost of an allocation A is defined as $C(A) = \sum_{j \in M} c_j(T_j)$. In Section 4.3, we also consider the *non-separable setting*, where we are given a more general cost function $C : (2^M)^n \to \mathbb{R}_{\geq 0}$, specifying for every allocation $A = (A_i)_{i \in N}$ the corresponding cost $C(A)$. Non-separable functions can capture dependencies among different items.

We assume that the utility functions of the players are *quasilinear*, i.e., given an allocation $A = (A_i)_{i \in N}$ and payments $(p_i)_{i \in N}$ determined by the mechanism for valuation functions $\vec{v} = (v_i)_{i \in N}$, the utility of player i is defined as $u_i(\vec{v}) = v_i(A_i) - p_i$. All our mechanisms have *no positive transfers* (NPT), i.e., $p_i \geq 0$, and satisfy *individual rationality* (IR), i.e., $p_i \leq v_i(A_i)$.

In addition to the above, we are also interested in the following properties:

- **Weak Group-Strategyproofness (WGSP):** We insist on a stronger notion of resistance to manipulation than truthfulness: A mechanism is *weakly group-strategyproof* if there is no deviation by a coalition of players that makes all its members strictly better off. More formally, we require that for every coalition $Q \subseteq N$ of players, every profile $\vec{v}_{-Q}$ of the other players, there is no deviation $\vec{b}_Q$ of the players in Q such that $u_i(\vec{b}_Q, \vec{v}_{-Q}) > u_i(\vec{v}_Q, \vec{v}_{-Q})$ for every $i \in Q$, where $\vec{v}_Q$ is the profile of the actual valuation functions of Q.

- **Budget Balance:** We are interested in mechanisms whose payments cover the allocation cost, ideally exactly. However, the latter is not always possible as it may be incompatible with the other objectives. We therefore consider an approximate budget balance notion: A mechanism is *β-budget-balanced* ($\beta \geq 1$) if for every valuation profile $\vec{v} = (v_i)_{i \in N}$, the outcome (A, p) computed by the mechanism satisfies

$$C(A) \leq \sum_{i \in N} p_i \leq \beta \cdot C(A).$$

  Clearly, we want β to be as small as possible to not overcharge players too much for covering the cost. We say that the mechanism is *budget balanced* if $\beta = 1$.

- **Economic Efficiency:** Our goal is to compute outcomes that are (approximately) efficient. To this aim, we use the *social cost objective*, originally defined in [17]. Adapted to our combinatorial setting, the *social cost* of an allocation $A = (A_i)_{i \in N}$ is defined as the actual cost of the outcome plus the value missed by not serving all items to all players, i.e.,

$$\pi(A) = \sum_{j \in M} c_j(T_j) + \sum_{i \in N} [v_i(M) - v_i(S_i)].^3$$

A mechanism is said to be α-*approximate* with respect to the social cost objective if for every valuation profile $\vec{v} = (v_i)_{i \in N}$, the allocation A output by the mechanism satisfies $\pi(A) \leq \alpha \cdot \pi(A^*)$, where A^* is an allocation of minimum social cost.

We assume that both the valuation functions $(v_i)_{i \in N}$ and the cost functions $(c_j)_{j \in M}$ are non-decreasing (see below for formal definitions). Further, we focus on certain classes of valuation and cost functions: More specifically, we consider *submodular* and *subadditive* cost functions, both naturally modeling economies of scale. As to the valuation functions, we consider *submodular* valuation functions in Section 3 and general valuation functions in Section 4. Further, the class of *symmetric XOS* functions play a prominent role in Section 3.[4] Below we summarize all relevant definitions (see also Lehman et al. [12]).

▶ **Definition 1.** *Let* $f : 2^U \to \mathbb{R}_{\geq 0}$ *be a function defined over subsets of a universe* U.
1. f *is* non-decreasing *if* $f(S) \leq f(T)$ *for every* $S \subseteq T \subseteq U$.
2. f *is* symmetric *if* $f(S) = f(T)$ *for every* $S, T \subseteq U$ *with* $|S| = |T|$.
3. f *is* submodular *if* $f(S \cup \{i\}) - f(S) \geq f(T \cup \{i\}) - f(T)$ *for every* $S \subseteq T \subseteq U$ *and* $i \notin S$.
4. f *is* XOS *if there are additive functions* $f^1, \ldots, f^k$ *such that* $f(S) = \max_{i \in [k]} f^i(S)$ *for all* $S \subseteq U$.
5. f *is* subadditive *if* $f(S \cup T) \leq f(S) + f(T)$ *for every* $S, T \subseteq U$.
6. f *is* symmetric XOS *if it is symmetric and* $f(S)/|S| \geq f(T)/|T|$ *for every* $S \subseteq T \subseteq U$.

Some of our mechanisms make use of cross-monotonic cost-sharing functions defined as follows:

▶ **Definition 2.** *Let* $c : 2^N \to \mathbb{R}_{\geq 0}$ *be a cost function. A* cost-sharing function[5] $\chi : N \times 2^N \to \mathbb{R}_{\geq 0}$ *with respect to* c *specifies for each subset* $S \subseteq N$ *and every player* $i \in S$ *a non-negative cost share* $\chi(i, S)$ *such that* $\sum_{i \in S} \chi(i, S) \geq c(S)$.[6] χ *is* cross-monotonic *if for all* $S \subseteq T \subseteq N$ *and every* $i \in S$, *we have* $\chi(i, S) \geq \chi(i, T)$.

3   An Iterative Ascending Cost Sharing Mechanism

In this section, we present our *Iterative Ascending Cost Sharing Mechanism (IACSM)* for the combinatorial cost sharing setting with symmetric submodular valuations and general cost functions. We first provide a generic description of our mechanism and identify two properties which are sufficient for our main result to go through. We then show that these properties are satisfied if the valuations are symmetric submodular.

3.1   Definition of IACSM and Two Crucial Properties

Mechanism IACSM can be viewed as a generalization of the *Moulin-Shenker* mechanism [15] to the combinatorial setting in the sense that it simulates in parallel an ascending iterative auction for each item. To our knowledge this is the first ascending price mechanism for the combinatorial setting which is not VCG-based and as we will describe below, this adaptation is not straightforward since there are several obstacles we need to overcome. A description of our mechanism IACSM is given in Algorithm 1.

---

[4] It is not hard to verify that these functions can equivalently be defined as stated in Definition 1 (see also [8]).
[5] We stress here that we allow cost-sharing functions to overcharge the actual cost $c(\cdot)$. As a result, this will lead to approximately budget balanced mechanisms.
[6] For notational convenience, we define $\chi(i, S) = \infty$ for $i \notin S$.

■ **Algorithm 1** Iterative Ascending Cost Sharing Mechanism (IACSM).

Input: Declared valuation functions $(b_i)_{i \in N}$.
Output: Allocation $A = (A_i)_{i \in N}$ and payments $p = (p_i)_{i \in N}$.

1 **Initialization:** Let $X = N$ be the set of active players and define $T_j = N$ for every item $j \in M$.

2 **while** $X \neq \emptyset$ **do**

3   Compute an *optimal bundle* A_i for every player $i \in X$:
$$A_i \in \arg\max_{S \subseteq M}\{b_i(S) - p_i(S)\}, \quad \text{where} \quad p_i(S) = \sum_{j \in S} \chi_j(T_j) \tag{1}$$

     (If there are several optimal bundles, resolve ties as described within Section 3.1.)

4   Let $i^* \in X$ be a player such that $|A_{i^*}| \leq |A_i|$ for every $i \in X$.

5   Assign the items in A_{i^*} to player i^* and remove player i^* from X.

6   For every item $j \in M \setminus A_{i^*}$, set $T_j = T_j \setminus \{i^*\}$, and update the cost shares $\chi_j(T_j)$.

7 **return** $A = (A_i)_{i \in N}$ and $p = (p_i)_{i \in N}$, where $p_i = \sum_{j \in A_i} \chi_j(T_j)$.

The mechanism maintains a set of *active* players X and for each item $j \in M$ a set of players T_j who are *tentatively* assigned to j. Initially, each player is active and tentatively assigned to all the items, i.e., $X = N$ and $T_j = N$ for all $j \in M$. The mechanism then proceeds in iterations. In each iteration, each item j is offered to each active player $i \in X$ at a price that only depends on the set of tentatively assigned players T_j. For this, we use a player-independent *cost sharing function* $\chi_j(\cdot, T_j)$ for every item j, and since we require that $\chi_j(i, T_j) = \chi_j(k, T_j)$ for every $i, k \in T_j$, we will simply denote by $\chi_j(T_j)$ the cost share that each player $i \in T_j$ tentatively assigned to j has to pay. Based on these cost shares, every active player $i \in X$ computes an *optimal bundle* A_i with respect to the payments $p_i(\cdot)$, as defined in Equation (1). If there are ties, we resolve them according to the following *tie-breaking rule*: if there are several optimal bundles, then player i chooses one of maximum size. If there are multiple optimal bundles of maximum size k, then she chooses the bundle consisting of the k cheapest items (where ties between equal cost share items are resolved consistently, say by index of the items).

After determining the optimal bundle for each active player, the mechanism then chooses an active player i^* whose optimal bundle has minimum size. Again, we break ties consistently, say by index of the players. The items in A_{i^*} are assigned to player i^* and i^* becomes inactive. Finally, for every item j that does not belong to the optimal bundle A_{i^*}, i^* is removed from the tentative set T_j. The mechanism terminates when all players are inactive.

We next identify two crucial properties that our mechanism has to satisfy for our main result to go through. To formalize these properties, we introduce first some more notation.

Trace of IACSM. Note that the execution of our mechanism IACSM on an instance of the problem induces an order $\tau = (\tau_1, \ldots, \tau_n)$ on the players. Without loss of generality, we may assume that the players are renamed such that $\tau = (1, \ldots, n)$, i.e., player i becomes inactive in iteration i; however, we emphasize that this order is determined by the run of our mechanism.

The order $\tau = (1, \ldots, n)$ together with the final bundle A_i assigned to each player i at the end of iteration i induces an order of player withdrawals for each item j. More precisely, for every $j \in M$ we let τ_j be the subsequence of τ consisting only of the players who withdrew from item j (at the end of the iteration when they became inactive). We refer to τ_j as the *trace of item j*. Recall that initially $T_j = N$ and hence all players are tentatively assigned to j. The length of τ_j can vary from 0, when nobody withdraws from item j and τ_j is the null sequence, all the way to n, when everybody withdraws from j and $\tau_j = \tau$. Given a trace τ_j

in the form $\tau_j = (i_1, i_2, \ldots, i_\ell)$ and $k \in \{0, 1, \ldots, |\tau_j|\}$, let $R_j^k = N \setminus \{i_1, i_2, \ldots, i_k\}$; define $R_j^0 = N$. Note that the set R_j^k is precisely the set of players tentatively assigned to j after k players have withdrawn from j during the execution of the mechanism. We note that the notion of trace is valid also for any other iterative mechanism where the assignment of one player becomes finalized at each iteration, e.g., [13].

Trace-monotonic cost sharing functions. We introduce a new property of cost sharing functions which will turn out to be crucial below. Intuitively, it is a refinement of the standard cross-monotonicity property which has to hold only for *certain* subsets of players encountered by the mechanism, namely for the sets $\{R_j^k\}$. More precisely, given a trace τ_j for an item $j \in M$, we say that the cost sharing function χ_j is *cross-monotonic with respect to τ_j* (or, *trace-monotonic* for short), if for every $k \in \{0, \ldots, |\tau_j| - 1\}$, we have $\chi_j(R_j^k) \leq \chi_j(R_j^{k+1})$. Note that this ensures that the cost share of item j (weakly) increases during the execution of the mechanism, as we consider the sequence of sets $R_j^0 \supset R_j^1 \supset \cdots \supset R_j^{|\tau_j|}$. A subtle point here is that the definition of the cost share $\chi_j(R_j^k)$ may not only depend on the set of players R_j^k, but also on the trace τ_j specifying how the set R_j^k has been reached by the mechanism.[7] It will become clear below that this additional flexibility enables us to implement our mechanism for *arbitrary* cost functions.

Properties (P1) and (P2). Our first property is rather intuitive: An item j needs to be offered to all active players at the same price and this price can only increase in subsequent iterations. In particular, this ensures that if at the end of iteration i, player i withdraws from an item $j \in M \setminus A_i$, then the price of j for the remaining players in $T_j \setminus \{i\}$ does not decrease. This is crucial to achieve strategyproofness, and it is captured precisely by trace-monotonic cost sharing functions.[8]

(P1) For each item $j \in M$ the cost sharing function χ_j is trace monotonic for every trace τ_j.

The first property alone is not sufficient to ensure that our mechanism IACSM is weakly group-strategyproof (or even strategyproof). Additionally, we need to enforce the following refinement property on the final bundles assigned to the players. We prove below that Property (P2) is satisfied for symmetric submodular valuation functions.

(P2) The final bundles $(A_i)_{i \in N}$ assigned to the players satisfy the following *refinement property*: $A_i \subseteq A_{i+1}$ for every $i \in \{1, \ldots, n-1\}$.

Feasibility of (P1) and (P2). We next define the cost sharing function that we use. The intuition is as follows: Suppose that $S = T_j$ is the set of players who are tentatively allocated to item j at the beginning of iteration i for some $i \in [n]$. Ideally, we would like to charge the average cost $c_j(S)/|S|$ to each player in S, but we cannot simply do this because the average cost might decrease with respect to the previous iteration, and this will destroy Property (P1). Given our new notion of trace-monotonicity, we can resolve this by defining the cost share of item j as the maximum average cost over all player sets which were tentatively allocated to j so far.

---

[7]   Notationally, we would have to write here $\chi_j^{\tau_j}$ to indicate the dependency on τ_j. However, in the analysis we focus on a fixed trace produced by an execution of the mechanism and omit the explicit reference to it for notational convenience.

[8]   Note that we have to require trace-monotonicity with respect to an *arbitrary* trace of item j here, because we cannot control the trace τ_j that will be realized by IACSM.

More formally, let τ_j be the trace of item j induced by IACSM when executed on a given instance. Let S be the set of players tentatively assigned to item j at the beginning of iteration i, and fix k such that $R_j^k = S$ (by the definition of our mechanism, such a k must exist and $k \leq i - 1$). We define

$$\chi_j(S) = \max_{\ell \in \{0, \ldots, k\}} \frac{c_j(R_j^\ell)}{|n - \ell|}. \tag{2}$$

Note that by using this definition we may end up overcharging the actual cost $c_j(S)$ of item j in the sense that $|S| \cdot \chi_j(S) > c_j(S)$. As we show in Section 3.2, the budget balance and social cost approximation guarantees depend on the magnitude by which we might overcharge.

It is now trivial to show that Property (P1) holds.

▶ **Lemma 3.** *Consider some item $j \in M$ and let $c_j : 2^N \to \mathbb{R}_{\geq 0}$ be an arbitrary cost function. Let τ_j be an arbitrary trace of j. The cost sharing function χ_j defined in (2) is trace-monotonic.*

We turn to Property (P2). In general, it seems difficult to guarantee (P2), but it is not hard to see that it holds if the valuation functions are symmetric submodular.

▶ **Lemma 4.** *Suppose the valuation functions are symmetric submodular. Then $A_i \subseteq A_{i+1}$ for every $i \in \{1, \ldots, n - 1\}$.*

3.2 Main result for IACSM

In order to state our main result of this section, we need to introduce a crucial parameter that determines the budget balance and social cost approximation guarantees of our mechanism.

▶ **Definition 5.** *A cost function $c : 2^N \to \mathbb{R}_{\geq 0}$ is α-average-decreasing, for some $\alpha \geq 1$, if for every $S \subseteq T \subseteq N$, $\alpha \cdot \frac{c(S)}{|S|} \geq \frac{c(T)}{|T|}$.*

Note that for every cost function c there exists some $\alpha \geq 1$ such that c is α-average decreasing. However, here we are particularly interested in α-average decreasing cost functions for which the parameter α is small, as can be seen by Theorem 6 below. Average decreasing functions with small values of α arise naturally in the domains of digital goods and public goods. For digital goods the cost of serving a non-empty set of customers is typically assumed to be constant because there is a cost for producing the good and then it can be shared with no additional cost (hence the definition is satisfied with $\alpha = 1$). The same is applicable for some public good models. Note also that *symmetric XOS cost functions* (see Definition 1) are average-decreasing (i.e., $\alpha = 1$).

The following is the main result of this section.

▶ **Theorem 6.** *Suppose the valuation functions are symmetric submodular and the cost functions are α-average decreasing, for some $\alpha \geq 1$. Then the mechanism IACSM runs in polynomial time, satisfies IR, NPT, WGSP and is α-budget balanced and $2\alpha^3 H_n$-approximate.*

Symmetric submodular cost functions are average decreasing (i.e., $\alpha = 1$) since they are a subclass of symmetric XOS functions. As a consequence, we obtain the following corollary from Theorem 6 (with an additional improvement on the social cost approximation).

▶ **Corollary 7.** *Suppose the valuation functions and the cost functions are symmetric submodular. Then the mechanism IACSM runs in polynomial time, satisfies IR, NPT, WGSP and is budget balanced and H_n-approximate.*

Note that the approximation factor of H_n for symmetric submodular functions is tight: The impossibility result of Dobzinski et al. [6] for a single public good implies that achieving a better approximation ratio is impossible, even in the single-item case ($m = 1$).

Finally we point out that *α-average-decreasing* functions are subadditive when $\alpha = 1$, while this is not necessarily true for $\alpha > 1$.

▶ **Lemma 8.** *Let $c(\cdot)$ be an α-average-decreasing cost function where $\alpha = 1$. Then $c(\cdot)$ is subadditive and in addition, not necessarily symmetric, or submodular. In case $c(\cdot)$ is α-average-decreasing with $\alpha > 1$, then $c(\cdot)$ is not necessarily subadditive.*

3.3 Proof of Social Cost Approximation

Due to lack of space, we will only establish the social cost approximation stated in Theorem 6. In this section, we will show that our mechanism IACSM is $2\alpha^3 H_n$-approximate with respect to the social cost objective for symmetric submodular valuation functions.

Let $A = (A_i)_{i \in N}$ be the allocation computed by the mechanism, where $A_i \subseteq M$ is the subset of items that player i receives. As before, without loss of generality we assume that the player order induced by IACSM is $\tau = (1, \ldots, n)$. Recall that for every item $j \in M$, $T_j = \{i \in N : j \in A_i\}$ is the final set of players that receive item j. We also use T_j^i to refer to the subset of players who are allocated to item j at the beginning of iteration i. Clearly, $T_j^i \supseteq T_j$ for every player i and item j.

We first state some simple lemmas which will be helpful to establish the social cost approximation guarantee.

▶ **Lemma 9.** *Fix an item $j \in M$ and let i be the first player in τ such that $j \in A_i$. Then $T_j = \{i, \ldots, n\}$.*

▶ **Lemma 10.** *Consider player i who becomes inactive in iteration i. We have*

$$v_i(A_i) - \sum_{j \in A_i} \chi_j(T_j) \geq v_i(S) - \sum_{j \in S} \chi_j(T_j) \quad \forall S \subseteq M.$$

Proof. In iteration i, the final bundle A_i is chosen as the set of items maximizing the utility of player i with respect to the current cost shares, i.e.,

$$v_i(A_i) - \sum_{j \in A_i} \chi_j(T_j^i) \geq v_i(S) - \sum_{j \in S} \chi_j(T_j^i) \quad \forall S \subseteq M. \tag{3}$$

Recall that T_j^i is the set of players that are allocated to item j in iteration i. Note that by Lemma 9, $T_j^i = T_j$ for every $j \in A_i$. Further, $T_j^i \supseteq T_j$ for every $j \in M \setminus A_i$ as additional players might withdraw from j in subsequent iterations. Note that the final set T_j is reached from T_j^i by following the trace τ_j of item j. The claim now follows from the trace-monotonicity of χ_j (Property (P1)). ◀

▶ **Lemma 11.** *Consider player i who becomes inactive in iteration i. For every item $j \in M$,*

$$\chi_j(T_j^i) \leq \alpha \frac{c_j(\{i, \ldots, n\})}{n - i + 1}.$$

▶ **Lemma 12.** *Let c be an α-average decreasing cost function. Let $S, T \subseteq N$ be arbitrary subsets with $|S| \leq |T|$. Then $c(S) \leq 2\alpha c(T)$.*

We are now ready to prove the approximation guarantee.

▶ **Lemma 13.** *Mechanisms IACSM is $2\alpha^3 H_n$-approximate.*

Proof. Let $A^* = (A_1^*, \ldots, A_n^*)$ be an optimal allocation and let T_j^* be the respective set of players that receive item j in A^*. We have

$$
\begin{aligned}
\pi(A) &= \sum_{i \in N} \big(v_i(M) - v_i(A_i)\big) + \sum_{j \in M} c_j(T_j) \\
&\leq \sum_{i \in N} v_i(M) - \sum_{i \in N} \bigg(v_i(A_i) - \sum_{j \in A_i} \chi_j(T_j)\bigg) \\
&\leq \sum_{i \in N} v_i(M) - \sum_{i \in N} \bigg(v_i(A_i^*) - \sum_{j \in A_i^*} \chi_j(T_j^i)\bigg) \\
&= \sum_{i \in N} \big(v_i(M) - v_i(A_i^*)\big) + \sum_{i \in N} \sum_{j \in A_i^*} \chi_j(T_j^i),
\end{aligned}
$$

where the first inequality holds because χ_j is α-budget balanced and the second inequality follows from Equation (3) in the proof of Lemma 10.

The proof follows if we can show that

$$
\sum_{i \in N} \sum_{j \in A_i^*} \chi_j(T_j^i) \leq 2\alpha^3 H_n \sum_{j \in M} c_j(T_j^*). \tag{4}
$$

We use a charging argument to prove (4). Fix some item $j \in M$ and order the players in T_j^* according to the player order $\tau = (1, \ldots, n)$ induced by IACSM; let $T_j^* = \{i_1, \ldots, i_{k_j^*}\}$ be the ordered set with $k_j^* := |T_j^*|$. We now "tag" each player i in T_j^* with a fraction of the cost $c_j(T_j^*)$ for item j as follows: For the lth player $i = i_l$ in T_j^* with $1 \leq l \leq k_j^*$, define

$$
\mathsf{tag}_i(j) := \frac{c_j(T_j^*)}{k_j^* - l + 1}. \tag{5}
$$

That is, the first player i_1 in T_j^* is tagged with $c_j(T_j^*)/k_j^*$, the second player i_2 with $c_j(T_j^*)/(k_j^* - 1)$ and so forth, and the last player $i_{k_j^*}$ is tagged with $c_j(T_j^*)$.

We first derive two lower bounds on the tagged cost:

▷ **Claim 14.** For every player $i \in N$ and for every item $j \in A_i^*$:

$$
\mathsf{tag}_i(j) \geq \frac{c_j(T_j^*)}{n - i + 1} \qquad \text{and} \qquad \mathsf{tag}_i(j) \geq \frac{c_j(T_j^*)}{|T_j^*|}.
$$

Proof. The latter bound holds by definition (5). To see that the former bound holds, observe that the kth last player ($1 \leq k \leq k_j^*$) in the ordered set T_j^* is tagged by $c_j(T_j^*)/k$. The claim now follows because there are at most $n - i$ players succeeding i in T_j^* according to the order. ◁

Note that the total tagged cost of item j satisfies

$$
\sum_{i \in T_j^*} \mathsf{tag}_i(j) = \sum_{l=1}^{k_j^*} \frac{c_j(T_j^*)}{k_j^* - l + 1} \leq H_n c_j(T_j^*). \tag{6}
$$

Thus, to prove (4) it suffices to show that the total cost share sum is upper bounded by the total tagged cost, i.e.,

$$
\sum_{i \in N} \sum_{j \in A_i^*} \chi_j(T_j^i) \leq 2\alpha^3 \sum_{j \in M} \sum_{i \in T_j^*} \mathsf{tag}_i(j). \tag{7}
$$

We show that for every i and every $j \in A_i^*$, $\chi_j(T_j^i) \leq \mathsf{tag}_i(j)$. Summing over all $i \in N$ and $j \in A_i^*$ then proves (4). We distinguish two cases:

Case 1: $|T_j^*| \geq n - i + 1$: Let $S \subseteq T_j^*$ be a set such that $|S| = n - i + 1$. We have

$$\chi_j(T_j^i) \leq \alpha \frac{c_j(\{i,\ldots,n\})}{n-i+1} \leq 2\alpha^2 \frac{c_j(S)}{|S|} \leq 2\alpha^2 \frac{c_j(T_j^*)}{n-i+1} \leq 2\alpha^2 \mathsf{tag}_i(j), \tag{8}$$

where the first inequality follows from Lemma 11, the second inequality follows from Lemma 12, the third inequality holds because c_j is non-decreasing and the last inequality follows from Claim 14.

Case 2: $|T_j^*| < n - i + 1$: Let $S \supset T_j^*$ be a set such that $|S| = n - i + 1$. We have

$$\chi_j(T_j^i) \leq \alpha \frac{c_j(\{i,\ldots,n\})}{n-i+1} \leq 2\alpha^2 \frac{c_j(S)}{|S|} \leq 2\alpha^3 \frac{c_j(T_j^*)}{|T_j^*|} \leq 2\alpha^3 \mathsf{tag}_i(j), \tag{9}$$

where the first inequality follows from Lemma 11, the second inequality follows from Lemma 12, the third inequality holds because c_j is α-average-decreasing and the last inequality follows from Claim 14. This concludes the proof. ◀

4 Mechanisms for General Valuations and Subadditive Cost Functions

In this section, we move away from symmetric submodular valuation functions and derive results for more general functions. In particular, we investigate the performance of the *Sequential Mechanism* [14] for general valuations and subadditive cost functions. Although for arbitrary subadditive cost functions this mechanism does not provide favorable approximation guarantees, we identify conditions on the cost functions under which it achieves significantly better approximation factors. This is based on considering a different parameterization of cost functions with regard to their average cost shares.

4.1 The Sequential Mechanism

The *Sequential Mechanism (SM)* was introduced by Moulin [14] and was also studied in [7]. A description of the mechanism SM is given in Algorithm 2. We note that this mechanism is applicable both to separable and non-separable cost functions. Here, we first focus on separable cost functions, and in Section 4.3, we consider generalizations to the non-separable setting.

It is trivial to see that SM is budget-balanced and it is also known that it is WGSP [7]. However, for arbitrary monotone subadditive cost functions, the mechanism achieves a (poor) social cost approximation guarantee of n (see [7]). Despite this, we show that SM has better guarantees under certain conditions. Namely, we identify a crucial parameter of each cost function c_j with $j \in M$ that allows us to quantify this improvement. The parameterization introduced here is different from the one used in Section 3 and it compares the average cost of a set $T \subseteq N$ with the minimum standalone cost of a player in T.

▶ **Definition 15.** *A cost function* $c : 2^N \to \mathbb{R}_{\geq 0}$ *is* α-*average min-bounded, for some* $\alpha \geq 1$, *if for every set* $T \subseteq N$, *we have* $\alpha \cdot \frac{c(T)}{|T|} \geq c_{\min}(T)$, *where* $c_{\min}(T) = \min_{j \in T} c(\{j\})$.

Definition 15 may look contrived at first glance and we thus provide some more intuition on how we arrived at this parameterization. Given that IACSM performs well for α-average decreasing functions and small values of α, as we established in Section 3, it is natural to

■ **Algorithm 2** Sequential Mechanism (SM).

Input: Declared valuation functions $(b_i)_{i \in N}$.
Output: Allocation $A = (A_i)_{i \in N}$ and payments $p = (p_i)_{i \in N}$.
1 **Initialization:** Fix an order on the set of players $N = \{1, \ldots, n\}$.
2 **for** $i = 1, \ldots, n$ **do**
3 $\quad$ Compute an *optimal bundle* A_i for player i:

$$A_i \in \arg\max_{S \subseteq M}\{b_i(S) - p_i(S)\}, \quad \text{where}$$

$$p_i(S) = C(A_1, \ldots, A_{i-1}, S, \emptyset, \ldots, \emptyset) - C(A_1, \ldots, A_{i-1}, \emptyset, \ldots, \emptyset).$$

$\quad$ (If there are multiple optimal bundles, choose the lexicographically smallest one.)

4 **return** $A = (A_i)_{i \in N}$ and $p = (p_i)_{i \in N}$, where $p_i = p_i(A_i)$.

---

focus on the complement of this class. For example, fix $\alpha = 1$ for now. Then the exact complement is not easy to characterize because it involves two existential quantifiers. We therefore consider a subset of this complement (with only one existential quantifier) by demanding that for every T, there exists $S \subseteq T$ such that $c(S)/|S| < c(T)/|T|$. It is not hard to verify that this definition is equivalent to the class of 1-average min-bounded functions. For larger values of α, we can see that α-average-min-bounded functions still capture a chunk of the complement of α-average-decreasing functions. Thus, a positive result for α-average-min-bounded functions narrows down on the cost functions that are not yet known to admit good approximation guarantees.

Note that for every cost function we can find an $\alpha \geq 1$ such that it is α-average min-bounded. As the next theorem reveals, the Sequential Mechanism attains a favorable performance for small values of α.

▶ **Theorem 16.** *Suppose we have general valuation functions and for each item $j \in M$, the cost function $c_j : 2^N \to \mathbb{R}_{\geq 0}$ is non-decreasing, subadditive, and α-average min-bounded for some $\alpha \geq 1$. Then the Sequential Mechanism satisfies IR, NPT, WGSP, and is budget balanced and $\alpha \cdot H_n$-approximate.*

For the proof of Theorem 16, we will use the following proposition:

▶ **Proposition 17.** *If $c : 2^N \to \mathbb{R}_{\geq 0}$ is non-decreasing and α-average min-bounded, then $\sum_{i \in T} c(\{i\}) \leq \alpha H_{|T|} \cdot c(T)$ for every $T \subseteq N$.*

Proof of Theorem 16. We only need to prove that SM is αH_n-approximate. All the other properties have been established in [7, 14]. Let $A = (A_i)_{i \in N}$ be the allocation output by the mechanism and let $A^* = (A_i^*)_{i \in N}$ be an optimal allocation. Further, let T_j^* be the respective set of players that receive item j in A^*. To simplify notation in the analysis, we also let $A_{<i}$ denote the tuple $(A_1, \ldots, A_{i-1}, \emptyset, \ldots, \emptyset)$. Define now the incremental cost of a player i for a bundle $S \subseteq M$, with respect to the allocation constructed by the Sequential Mechanism before i's turn as $\Delta_i(A_{<i}, S) = C(A_1, \ldots, A_{i-1}, S, \emptyset, \ldots, \emptyset) - C(A_1, \ldots, A_{i-1}, \emptyset, \ldots, \emptyset)$.
We have

$$\pi(A) = \sum_{i \in N} \left[v_i(M) - v_i(A_i)\right] + C(A) = \sum_{i \in N} v_i(M) - \sum_{i \in N} \left[v_i(A_i) - \Delta_i(A_{<i}, A_i)\right]$$

$$\leq \sum_{i \in N} \left[v_i(M) - v_i(A_i^*)\right] + \sum_{i \in N} \Delta_i(A_{<i}, A_i^*).$$

Note that the inequality holds because A_i was chosen as the optimal bundle for i. The next step is to prove a bound on the incremental costs in the form

$$\sum_{i \in N} \Delta_i(A_{<i}, A_i^*) \leq \beta \cdot C(A^*). \tag{10}$$

The proof follows if we can show that (10) holds for $\beta = \alpha H_n$ because we then have

$$\pi(A) \leq \sum_{i \in N} \left[v_i(M) - v_i(A_i^*) \right] + \alpha \cdot H_n C(A^*) \leq \alpha H_n \cdot \pi(A^*).$$

By exploiting the subadditivity of the cost functions c_j, we obtain

$$\Delta_i(A_{<i}, A_i^*) = C(A_{<i}, A_i^*) - C(A_{<i}) \leq C(A_{<i}) + C(A_i^*, \emptyset_{-i}) - C(A_{<i}) = \sum_{j \in A_i^*} c_j(\{i\}).$$

Summing over over all $i \in N$, and using Proposition 17, we get:

$$\sum_{i \in N} \Delta_i(A_{<i}, A_i^*) \leq \sum_{i \in N} \sum_{j \in A_i^*} c_j(\{i\}) = \sum_{j \in M} \sum_{i \in T_j^*} c_j(\{i\}) \leq \sum_{j \in M} \alpha H_{|T_j^*|} c_j(T_j^*) \leq \alpha H_n C(A^*).$$

This proves (10) and concludes the proof of Theorem 16.   ◄

By going through the proof of Theorem 16 more carefully, we realize the following:

▶ **Remark 18.** For any subclass of non-decreasing, subadditive cost functions, it suffices to establish inequality (10) to prove that the Sequential Mechanism has a social cost approximation guarantee of β.

Finally we have that for $\alpha = 1$ the approximation factor is tight.

▶ **Proposition 19.** *Even for the single item setting, there exists a 1-average min-bounded cost function, under which the Sequential Mechanism provides an H_n-approximation.*

4.2   Improved Approximation Guarantees and Applications

We continue with a natural refinement of Definition 15 which turns out to provide even better approximation factors of the Sequential Mechanism.

▶ **Definition 20.** *A cost function $c : 2^N \to \mathbb{R}_{\geq 0}$ is α-average max-bounded, for some $\alpha \geq 1$, if for every set $T \subseteq N$, we have $\alpha \cdot \frac{c(T)}{|T|} \geq c_{\max}(T)$, where $c_{\max}(T) = \max_{j \in T} c(\{j\})$.*

Clearly, any function that is α-average max-bounded is also α-average min-bounded. Thus, we already have an αH_n-approximation for non-decreasing, subadditive and α-average max-bounded cost functions. Below we show that we can achieve a much better guarantee.

▶ **Theorem 21.** *Suppose we have general valuation functions and for each item $j \in M$, the cost function $c_j : 2^N \to \mathbb{R}_{\geq 0}$ is non-decreasing, subadditive, and α-average max-bounded for some $\alpha \geq 1$. Then the Sequential Mechanism satisfies IR, NPT, WGSP, and is budget-balanced and α-approximate.*

Example applications of combinatorial cost functions. We give some examples of combinatorial cost functions below and show that they are α-average max-bounded (possibly depending on some parameters of the combinatorial problem). In particular, by applying Theorem 21 we obtain attractive social cost approximation guarantees for these problems. For simplicity, all examples consider a single item only; but clearly, we can consider more general multiple item settings (e.g., when for each item $j \in M$, c_j captures one of the problems below).

- **Set Cover.** We are given a universe of elements U and a family $\mathcal{F} \subseteq 2^U$ of subsets of U. The players correspond to the elements of U and the cost $c(S)$ for serving a set of players $S \subseteq U$ is defined as the size of a minimum cardinality set cover for S.
- **Vertex Cover.** This is a special case of Set Cover. We are given an undirected and unweighted graph $G = (V, E)$ and the players are the edges of the graph. The cost $c(S)$ for serving a set $S \subseteq E$ of players is defined as the size of a minimum vertex cover in the subgraph induced by S.
- **Matching.** We are given an undirected and unweighted graph $G = (V, E)$ and the players correspond to the edges. The cost $c(S)$ for serving a set S of players is defined as the size of a maximum cardinality matching in the subgraph induced by S.

Using our α-average max-bounded notion, it is now easy to prove that these problems admit constant social cost approximation guarantees (under certain restrictions).

▶ **Theorem 22.** *The Sequential Mechanism is α-approximate for the above problems, where*
1. *$\alpha = d$ for the Set Cover problem, where d is the maximum cardinality of the sets in $\mathcal{F}$;*
2. *$\alpha = k$ for the Vertex Cover problem in graphs of maximum degree k;*
3. *$\alpha = k$ for the Matching problem in bipartite graphs of maximum degree k;*
4. *$\alpha = (5k + 3)/4$ for the Matching problem in general graphs of maximum degree k.*

We can now compare these bounds with the existing results in the literature. For Vertex Cover, there is a mechanism that is 2-budget-balanced and $O(\log n)$-approximate [13]. Thus, for graphs with maximum degree less than $\log n$, we obtain a better guarantee. For Set Cover, there is a mechanism that is $O(\log n)$-budget-balanced and $O(\log n)$-approximate [13]. Hence, we obtain an improvement if the sets in $\mathcal{F}$ have a size that is no more than $o(\log n)$. Finally, we note that our results do not apply to the weighted versions of these problems.

4.3 Guarantees of the Sequential Mechanism for Non-Separable Cost Functions

We extend our results to non-separable cost functions. Recall that in this setting, the cost $C(A)$ of an allocation $A = (A_i)_{i \in N}$ is given by some general (not necessarily separable) cost function $C : (2^M)^n \to \mathbb{R}_{\geq 0}$. In particular, C may encode dependencies among different items.

We introduce some more notation. Given two allocations $S = (S_i)_{i \in N}$ and $T = (T_i)_{i \in N}$, we define $S \cup T$ as the componentwise union of S and T, i.e., $S \cup T = (S_1 \cup T_1, \ldots, S_n \cup T_n)$. Similarly, we write $S \subseteq T$ if this relation holds componentwise, i.e., $S_i \subseteq T_i$ for every $i \in N$. Given an allocation $A = (A_i)_{i \in N}$ and a set of players $S \subseteq N$, we define $A|_S = (A_S, \emptyset_{-S})$ as the allocation in which each player $i \in S$ receives the items in A_i and all other players receive nothing. If $S = \{i\}$ is a singleton set, we also write $A|_i$ instead of $A|_{\{i\}}$. Throughout this section, we remain in the domain of non-decreasing and subadditive cost functions. In the non-separable case, a cost function $C : (2^M)^n \to \mathbb{R}_{\geq 0}$ is non-decreasing if $C(S) \leq C(T)$ for every pair of allocations S, T, with $S \subseteq T$. Also, it is subadditive if for every two allocations $S = (S_i)_{i \in N}$ and $T = (T_i)_{i \in N}$, we have $C(S \cup T) \leq C(S) + C(T)$.

We now adapt Definitions 15 and 20 to non-separable cost functions.

▶ **Definition 23.** *Let $C : (2^M)^n \to \mathbb{R}_{\geq 0}$ be a non-separable cost function.*

- *C is α-average min-bounded, for some $\alpha \geq 1$, if for every allocation A and every subset $T \subseteq N$ with $|T| \geq 2$, it holds $\alpha \frac{C(A|_T)}{|T|} \geq C_{\min}(T)$, where $C_{\min}(T) = \min_{j \in T} C(A|_j)$.*
- *C is α-average max-bounded, for some $\alpha \geq 1$, if for every allocation A and every subset $T \subseteq N$ with $|T| \geq 2$, it holds $\alpha \frac{C(A|_T)}{|T|} \geq C_{\max}(T)$, where $C_{\max}(T) = \max_{j \in T} C(A|_j)$.*

As before, if a non-separable function is α-average max-bounded, then it is also α-average min-bounded.

We remark that it has been shown in [14, 7] that the Sequential Mechanism is weakly group-strategyproof and budget balanced for the non-separable setting. By adapting Proposition 17 for the non-separable setting and by using the same reasoning as in the proof of Theorem 16, we obtain the same approximation guarantee of αH_n as in the separable setting. Further, the improvement we obtained in Theorem 21 also goes through in this setting. We summarize these observations in the following corollary.

▶ **Corollary 24.** *Suppose we have general valuation functions and a non-decreasing, subadditive, and α-average min-bounded cost function $C : (2^M)^n \to \mathbb{R}_{\geq 0}$. Then the Sequential Mechanism satisfies IR, NPT, WGSP, and is budget balanced and $\alpha \cdot H_n$-approximate. Furthermore, if C is also α-average max-bounded, then the Sequential Mechanism is α-approximate.*

5   Discussion

In Section 3, we proposed the mechanism IACSM, which is weakly group-strategyproof under general cost functions and symmetric submodular valuations. Moreover it is α-budget balanced and $2\alpha^3 H_n$-approximate when we restrict the cost functions to the α-average-decreasing class. The social cost approximation guarantee further improves to H_n if the cost functions are symmetric submodular and this is best possible (due to the known lower bound for public-excludable goods [6]). It would be very interesting to explore mechanisms that go beyond symmetric submodular valuation functions. It seems that entirely new ideas are needed for this setting. It would also be interesting to extend our mechanism to non-separable cost functions. We note that separability of the costs in Section 3 is needed for IACSM only to argue that the cost share per item increases as players withdraw (with respect to the trace). One would need to investigate how to adapt the mechanism and enforce this property in the non-separable setting. Technically, this seems far from obvious and we leave a proper treatment of this issue for future work.

In Section 4, we studied the (partially) complementary class of α-average min bounded cost functions. We showed that the well-known Sequential Mechanism is budget balanced and αH_n-approximate even for general valuation functions. These results also extend to non-separable cost functions. A very natural question is whether SM is optimal in this setting and we note that the answer is not yet clear: The impossibility result of [6] holds for the public-excludable good cost function which is symmetric submodular and thus 1-average-decreasing. However, it is not hard to see that this does not fall within the α-average min bounded class for any constant α. This leads to the question of whether there exists a WGSP mechanism that breaks the $\Omega(\log(n))$-approximation in terms of social cost for α-average-min bounded functions with small values of α.

Finally, what we also find very interesting is to identify the class of cost functions for which neither of the two mechanisms studied here perform well. Recall that, for any constant value of α, if a cost function is either α-average decreasing or α-average min-bounded, then a good performance is guaranteed. Thus, we need look at the complement of the set of

α-average decreasing functions and the set of α-average min-bounded functions for small value of α and examine whether these complements have a non-empty intersection. The following proposition shows that this intersection is indeed non-empty.

▶ **Proposition 25.** *Given $\alpha \geq 1$, the intersection of the complements of α-average-decreasing and α-average min-bounded functions is non-empty.*

The proof of this proposition follows by constructing a cost function that requires non-constant values of α to be captured by either of our parameterizations. Although the intersection turns out to be non-empty, the constructed cost function is rather artificial and more natural examples are elusive so far. In fact, for most of the known cost functions that have been studied in the literature, at least one of our mechanisms achieves an $O(H_n)$-approximation. To make further progress, we believe it is important to understand better the class of functions defined by the intersection of the two complements, as it would help us to identify the missing elements for deriving mechanisms for a wider class of cost functions.

References

1 G. Birmpas, C. Courcoubetis, I. Giotis, and E. Markakis. Cost-Sharing Models in Participatory Sensing. In *International Symposium on Algorithmic Game Theory*, pages 43–56, 2015.

2 Y. Bleischwitz and F. Schoppmann. Group-Strategyproof Cost Sharing for Metric Fault Tolerant Facility Location. In *International Symposium on Algorithmic Game Theory*, pages 350–361, 2008.

3 J. A. Brenner and G. Schäfer. Cost Sharing Methods for Makespan and Completion Time Scheduling. In *Symposium on Theoretical Aspects of Computer Science*, pages 670–681, 2007.

4 S. Chawla, T. Roughgarden, and M. Sundararajan. Optimal Cost-Sharing Mechanisms for Steiner Forest Problems. In *International Workshop on Internet and Network Economics*, pages 112–123, 2006.

5 N. R. Devanur, M. Mihail, and V. V. Vazirani. Strategyproof cost-sharing mechanisms for set cover and facility location games. *Decision Support Systems*, 39(1):11–22, 2005.

6 S. Dobzinski, A. Mehta, T. Roughgarden, and M. Sundararajan. Is Shapley cost sharing optimal? *Games and Economic Behavior*, 108:130–138, 2018.

7 S. Dobzinski and S. Ovadia. Combinatorial Cost Sharing. In *ACM Conference on Economics and Computation*, pages 387–404, 2017.

8 T. Ezra, M. Feldman, T. Roughgarden, and W. Suksompong. Pricing Identical Items. *CoRR*, abs/1705.06623, 2017. `arXiv:1705.06623`.

9 J. Feigenbaum, A. Krishnamurthy, R. Sami, and S. Shenker. Hardness results for multicast cost sharing. *Theoretical Computer Science*, 1-3(304):215–236, 2003.

10 J. Green, E. Kohlberg, and J. J. Laffont. Partial Equilibrium Approach to the Free Rider Problem. *Journal of Public Economics*, 6:375–394, 1976.

11 A. Gupta, Jochen Könemann, Stefano Leonardi, R. Ravi, and Guido Schäfer. Efficient cost-sharing mechanisms for prize-collecting problems. *Math. Program.*, 152(1-2):147–188, 2015.

12 B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.

13 A. Mehta, T. Roughgarden, and M. Sundararajan. Beyond Moulin mechanisms. *Games and Economic Behavior*, 67(1):125–155, 2009.

14 H. Moulin. Incremental cost sharing: Characterization by coalition strategy-proofness. *Soc. Choice Welfare*, 16:279–320, 1999.

15 H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: Budget balance vs efficiency. *Economic Theory*, 18:511–533, 2001.

16 K. Roberts. The Characterization of Implementable Choice Rules. In J. J. Laffont, editor, *Aggregation and Revelation of Preferences*. Amsterdam: North Holland, 1979.

17 T. Roughgarden and M. Sundararajan. Quantifying inefficiency in cost-sharing mechanisms. *Journal of the ACM*, 56(4):23:1–23:33, 2009.