Fair Allocation of Indivisible Goods: Improvement and Generalization

Mohammad Ghodsi * MohammadTaghi HajiAghayi †‡ Masoud Seddighin *
Saeed Seddighin †‡ Hadi Yami †‡

"Being good is easy, what is difficult is being just." [36] (Victor Hugo, 1862)

Abstract

We study the problem of fair allocation for indivisible goods. We use the *the maxmin share* paradigm introduced by Budish [15] as a measure for fairness. Procaccia and Wang [46] (EC'14) were first to investigate this fundamental problem in the additive setting. In contrast to what real-world experiments suggest, they show that a maxmin guarantee (1-MMS allocation) is not always possible even when the number of agents is limited to 3. While the existence of an approximation solution (e.g. a 1/2-MMS allocation) is quite straightforward, improving the guarantee becomes subtler for larger constants. Procaccia and Wang [46][1] provide a proof for existence of a 2/3-MMS allocation and leave the question open for better guarantees.

Our main contribution is an answer to the above question. We improve the result of Procaccia and Wang to a 3/4 factor in the additive setting. The main idea for our 3/4-MMS allocation method is clustering the agents. To this end, we introduce three notions and techniques, namely *reducibility*, *matching allocation*, and *cycle-envy-freeness*, and prove the approximation guarantee of our algorithm via non-trivial applications of these techniques. Our analysis involves coloring and double counting arguments that might be of independent interest.

One major shortcoming of the current studies on fair allocation is the additivity assumption on the valuations. We alleviate this by extending our results to the case of submodular, fractionally subadditive, and subadditive settings. More precisely, we give constant approximation guarantees for submodular and XOS agents, and a logarithmic approximation for the case of subadditive agents. Furthermore, we complement our results by providing close upper bounds for each class of valuation functions. Finally, we present algorithms to find such allocations for additive, submodular, and XOS settings in polynomial time. The reader can find a summary of our results in Tables 1 and 2.

1 Introduction

Suppose we have a set of m indivisible items, and wish to distribute them among n agents. Agents have valuations for each set of items that are not necessarily identical. How hard is it to divide the items between the agents to make sure everyone receives a fair share?

*Sharif University of Technology
†University of Maryland
‡Supported in part by NSF CAREER award CCF-1053605, NSF BIGDATA grant IIS-1546108, NSF AF:Medium grant CCF-1161365, DARPA GRAPHS/AFOSR grant FA9550-12-1-0423, and another DARPA SIMPLEX grant. Portions of this research were completed while the first and the second authors were visitors at the Simons Institute for the Theory of Computing.
[1]Recipient of the best student paper award at EC'14.

Fair division problems have been vastly studied in the past 60 years, (see, e.g. [2, 4, 11, 15, 21, 46, 47, 1]). This line of research was initiated by the work of Steinhaus in 1948 [47] in which the author introduced the *cake cutting* problem as follows: given a heterogeneous cake and a set of agents with different valuation functions, the goal is to find a fair allocation of the cake to the agents.

In order to study this problem, several notions of fairness are proposed, the most famous of which are *proportionality* and *envy-freeness*, introduced by Steinhaus in 1948 [47] and Foley in 1967 [30]. A division is called proportional, if the total value of the allocated pieces to each agent is at least $1/n$ fraction of his total value for the entire cake, where n is the number of agents. In an envy-free division, no agent wishes to exchange his share with another agent, i.e., every agent's valuation for his share is at least as much as his valuation for the other agents' shares. Clearly, proportionality is implied by envy-freeness.

Dubins and Spainer in 1961 [21] propose a simple *moving knife* procedure that can guarantee a proportional division of the cake. For envy-freeness, Selfridge and Conway design an algorithm that guarantees envy-freeness when the number of agents is limited to 3. Later, Brams and Taylor extend this guarantee to an arbitrary number of agents in the additive setting [12].

The problem becomes even more subtle when we assume the items are indivisible. It is not hard to show that for indivisible items, neither of proportionality nor envy-freeness can be guaranteed; for instance, when the number of items is smaller than the number of agents, at least one agent receives no items.

From a theoretical standpoint, proportionality and envy-freeness are too strong to be delivered in the case of indivisible goods. Therefore, Budish [15] proposed a newer notion of fairness for indivisible goods, namely *the maxmin share*, which attracted a lot of attention in recent years [46, 2, 39, 11, 16, 7, 48, 23]. Imagine that we ask an agent a_i to partition a set $\mathcal{M}$ of m items into n bundles and collect the bundle with the smallest value. To maximize his profits, agent a_i tries to divide $\mathcal{M}$ in a way that maximizes the value of the bundle with the lowest value to him. Based on this, the maxmin share of an agent a_i, denoted by MMS_i, is the value of the least valuable bundle in agent a_i's allocation; that is, the maximum profit a_i can obtain in this procedure. Clearly, MMS_i is the most that can be guaranteed to an agent, since if all valuations are the same, at least one agent obtains a valuation of at most MMS_i from his allocated set. The question is then, whether MMS_i is a feasible guarantee? Therefore, we call an allocation MMS, if every agent a_i receives a collection of items that are together worth at least MMS_i to him. Bouverret [11] showed that for the restricted cases, when the valuations of the items for each agent are either 0 or 1, or when $m \leq n + 3$, an MMS allocation is guaranteed to exist. In other words, each a_i can be guaranteed to receive a profit of at least MMS_i from his allocated items.

While the experiments support the existence of an MMS allocation in general [11], this theory was refuted by the pioneering work of Procaccia and Wang [46]. Procaccia and Wang [46] provided a surprising counter-example that admits no MMS allocation. They also show that a 2/3-MMS allocation always exists, i.e. there exists an algorithm that allocates the items to the agents in such a way that every agent a_i receives a share that is worth at least $2/3\mathsf{MMS}_i$ to him. In particular, they show for $n \leq 4$, their algorithm finds a 3/4-MMS allocation. However, their algorithm does not run in polynomial time unless we assume the number of agents is bounded by a constant number. In a recent work, Amanatidis, Markakis, Nikzad, and Saberi [2], improve this result by presenting a PTAS algorithm for finding a $(2/3 - \epsilon)$-MMS allocation to any number of agents. However, the heart of their algorithm is the same as [46]. In addition to this, Amanatidis *et al.* prove that for $n = 3$, a 7/8-MMS allocation is always possible. Note that, the counter example provided by Procaccia and Wang [46] requires a number of goods that is exponential to the number of agents. Kurokawa, Procaccia, and Wang in [39] provided a better construction for the counter-example

with a linear number of goods.

In this work, we improve the result of Procaccia and Wang [46] by proving that a 3/4-MMS allocation always exists. We also give a PTAS algorithm to find such an allocation in polynomial time. Of course, this only holds if the valuation of the agents for the items are additive. We further go beyond the additive setting and extend this result to the case of submodular, XOS, and subadditive settings. More precisely, we give constant approximation algorithms for submodular and XOS settings that run in polynomial time. For the subadditive case, we prove that a $1/10\lceil \log m \rceil$-MMS allocation is guaranteed to exist. We emphasize that finding the exact value of MMS_i for an agent is NP-hard. Furthermore, to the best of our knowledge, no PATS is known for computing the MMS values in non-additive settings. Thus, any α-MMS allocation algorithm in non-additive settings must overcome the difficulty that the value of MMS_i is not known in advance. Therefore, our algorithms don't immediately follow from our existential proofs.

In order to present the results and techniques, we briefly state the fair allocation problem. Note that you can find a formal definition of the problem with more details in Section 2. The input to a maxmin fair allocation problem is a set $\mathcal{M}$ of m items and a set $\mathcal{N}$ of n agents. Fix an agent $a_i \in \mathcal{N}$ and let $V_i : 2^{\mathcal{M}} \to \mathbb{R}^+$ be the valuation function of a_i. Consider the set Π_r of all partitionings of the items in $\mathcal{M}$ into r non-empty sets. We define $\mathsf{MMS}^r_{V_i}(\mathcal{M})$ as follows:

$$\mathsf{MMS}^r_{V_i}(\mathcal{M}) = \max_{P^*=\langle P^*_1, P^*_2, \ldots, P^*_r\rangle \in \Pi_r} \min_{1\le j\le r} V_i(P^*_j).$$

In the context of fair allocation, we denote the maxmin value of an agent a_i by $\mathsf{MMS}_i = \mathsf{MMS}^n_{V_i}(\mathcal{M})$. The fair allocation problem is defined as follows: *for a given parameter α, can we distribute the items among the agents in such a way that every agent a_i receives a set of items with a value of at least $\alpha\mathsf{MMS}_i$ to him?* Such an allocation is called an α-MMS allocation. We consider the fair allocation problem in both additive and non-additive settings (including submodular, XOS, and subadditive valuations). For non-additive settings, we use oracle queries to access the valuations. Note that, for non-additive settings, eliciting the exact valuation function of each agent needs an exponential number of queries. However, our methods for allocating the items in non-additive settings only uses a polynomial number of queries.

There are many applications for finding fair allocations in the additive and non-additive settings. For example, *spliddit*, a popular fair division website[2] suggests indisputable and provably fair solutions for many real-world problems such as sharing rents, distributing tasks, dividing goods, etc. For dividing goods, *spliddit* uses the maximum Nash welfare allocation (the allocation that maximizes the product of utilities). In [16] (EC'17), Caragiannis et.al., proved with a tight analysis that a maximum Nash welfare allocation is a $2/(1+\sqrt{4n-3})$-MMS allocation. However, the current best approximation gurantee and the state-of-the-art method for allocating indivisible goods is based on the result of [46] that guarantees a 2/3-MMS allocation. We believe our results can improve their performance. It is worth mentioning that despite the complexity of analysis, the idea behind our algorithm is simple and it can be easily implemented. The reader can find a set of materials including the implementation of our method and an animated explanation of our algorithm in https://www.cs.umd.edu/∼saeedrez/fair.html.

1.1 Relation to other Fundamental Problems

In this work, we study the allocation of indivisible items to maximize fairness. However, maximizing fairness is not the only goal that has been considered in the literature. In the following, we briefly

[2] `http://www.spliddit.org`

explain other variants of this problem that are very fundamental and have received a lot of attention in recent years.

Welfare maximization: Perhaps the simplest version of the resource allocation problem with indivisible goods is *the welfare maximization problem*. In this problem, we are given a set $\mathcal{M}$ of goods, and the goal is to allocate the items to a number of agents to maximize the welfare. The problem is trivial if we assume the agents to be additive. Therefore, the main focus has been on submodular, XOS, and subadditive agents [19, 20, 25, 41](STOC'05, SODA'06, STOC'06). Feige [25] gives tight algorithms that solve the problem in the subadditive and XOS settings with approximation factors $1/2$ and $1-1/e$, respectively. The approximation factors match the existing lower bounds for these settings.

Max-min allocation: Another variant of this problem is to maximize the least value that any agent obtains from the allocation. Asadpour and Saberi [4](STOC'07) give the first polynomial time algorithm for this problem that approximates the optimal solution within a factor of $O(\sqrt{n}\log^3 n)$ in the additive setting. This was later improved by Chakrabarty, Chuzhoy,and Khanna [17](FOCS'09) to an $O(m^\epsilon)$ approximation factor. This problem has also been studied with non-additive agents by Goemans *et al.* [33](SODA'09). They give an $O(\sqrt{mn}^{1/4}\log n\log^{3/2} m)$ approximation algorithm that runs in polynomial time and solves the problem when the agents valuations are submodular.

Santa Claus: A special case of the above problem in which the valuation of every agent for an item is either 0, or a fixed value is called *the Santa Claus problem*. This problem was first introduced by Bansal and Sviridenko [6] in STOC'09. In this paper, they give an $O(\log\log n/\log\log\log n)$ approximation algorithm for this problem that runs in polynomial time. Later Feige [24](SODA'08) showed that the objective value of the problem can be approximated within a constant factor in polynomial time. This was later turned into a constructive proof by Annamalai *et al.* [3](SODA'14).

1.2 Our Results and Techniques

Throughout this paper, we study the fair allocation problem for additive and non-additive agents. Procaccia and Wang [46] study the fair allocation problem and show a 2/3-MMS allocation is guaranteed to exist for any number of additive agents. We improve this result in two different dimensions: (i) we improve the factor $2/3$ to a factor $3/4$ for additive agents. (ii) we provide similar guarantees for submodular, fractionally subadditive, and subadditive agents. Moreover, we provide algorithms that find such allocations in polynomial time. A brief summary of our results is illustrated in Tables 1 and 2.

1.2.1 Additive Setting

While the existence of a 1/2-MMS allocation is trivial in additive setting (see the rest of this section for more details), obtaining a better bound is more complicated. As mentioned before, the pioneering work of Procaccia and Wang [46] presented the first proof to the existence of a 2/3-MMS allocation in the additive setting. On the negative side, they show that their analysis is tight, i.e. their method cannot be used to obtain a better approximation guarantee. However, whether or not a better bound could be achieved via a more efficient algorithm remains open as Procaccia and Wang [46] pose this question as an open problem.

We answer the above question in the affirmative. Our main contribution is a proof to the existence of a 3/4-MMS allocation for additive agents. Furthermore, we show that such an allocation can be found in polynomial time. This result improves the work of Procaccia and Wang [46] and Amanatidis *et al.* [2] where the former gives a proof to the existence of a 2/3-MMS allocation and the latter presents a PTAS algorithm for finding a 2/3-MMS allocation.

Table 1: Summary of the results

	Additive	Submodular	XOS	Subadditive
Previous work (existential proof)	2/3-MMS [46]	1/10-MMS [7][3]	-	-
Previous work (polytime algorithm)	$2/3-\epsilon$-MMS [2]	1/31-MMS [7]	-	-
Previous work (upper bound)	$1-\epsilon$-MMS [46]	-	-	-
Our results (existential proof)	3/4-MMS Theorem 3.39	1/3-MMS Theorem 4.7	1/5-MMS Theorem 5.4	$1/10\lceil\log m\rceil$-MMS Theorem 6.3
Our results (polytime algorithm)	$3/4-\epsilon$-MMS Theorem 3.40	1/3-MMS Theorem 4.8	1/8-MMS Theorem 5.5	-
Our results (upper bound)	-	3/4-MMS Theorem 4.2	1/2-MMS Theorem 4.1	1/2-MMS Theorem 4.1

Table 2: Results for a limited number of agents in the additive setting

	$n=3$	$n=4$
Procaccia and Wang [46]	3/4-MMS	3/4-MMS
Amanatidis *et al.* [2]	7/8-MMS	-
Our result	-	4/5-MMS Theorem A.6

Theorem 3.40 [restated]. *Any fair allocation problem with additive agents admits a* 3/4-MMS *allocation. Moreover, a* $(3/4-\epsilon)$-MMS *allocation can be found in time* $\mathsf{poly}(n,m)$ *for any* $\epsilon>0$.

The result of Theorem 3.40 is surprising, since most of the previous methods provided for proving the existence of a 2/3-MMS allocation were tight. This convinced many in the community that 2/3 is the best that can be guaranteed. This shows that the current techniques and known structural properties of maxmin share are not powerful enough to prove the bounds better than 2/3. In this paper, we provide a better understanding of this notion by demonstrating several new properties of maxmin share. For example, we introduce a generalized form of reducibility and develop double counting techniques that are closely related to the concept of maxmin-share.

For a better understanding of our algorithm, we start with the case where valuations of the agents for all items are small enough. More precisely, let $0<\alpha<1$ be a constant number and assume for every agent a_i and every item b_j, the value of agent a_i for item b_j is bounded by $\alpha\mathsf{MMS}_i$. In this case, we propose the following simple procedure to allocate the items to the agents.

- Arrange the items in an arbitrary order.
- Start with an empty bag and add the items to the bag one by one with respect to their order.
- Every time the valuation of an agent a_i for the set of items in the bag reaches $(1-\alpha)\mathsf{MMS}_i$, give all items of the bag to that agent, and continue with an empty bag. In case many agents

[3]In a parallel work to ours, Barman and Murthy in [7] (**EC'17**) considered the submodular case and proposed a 1/31 approximation guarantee.

are qualified to receive the items, we choose one of them arbitrarily. From this point on, we exclude the agent who received the items from the process.

We call this procedure the bag filling algorithm. One can see this algorithm as an extension of the famous moving knife algorithm for indivisible items. It is not hard to show that the bag filling algorithm guarantees a $(1-\alpha)$-MMS allocation to all of the agents. The crux of the argument is to show that every agent receives at least one bag of items. To this end, one could argue that every time a set of items is allocated to an agent a_i, no other agent a_j loses a value more than MMS_j. This, together with the fact that $V_i(\mathcal{M}) \geq n\mathsf{MMS}_i$, shows that at the end of the algorithm, every agent receives a fair share ($(1-\alpha)$-MMS) of the items.

This observation sheds light on the fact that low-value items can be distributed in a more efficient way. Therefore, the main hardness is to allocate the items with higher values to the agents. To overcome this hardness, we devise a clustering method. Roughly speaking, we divide the agents into three clusters according to their valuation functions. We prove desirable properties for the agents of each cluster. Finally, via a procedure that is similar in spirit to the bag filling algorithm but more complicated, we allocate the items to the agents.

Our clustering method is based on three important principles: *reducibility*, *matching allocation*, and *cycle-envy-freeness*. We give a brief description of each principle in the following.

Reducibility: The reducibility principle is very simple and elegant but plays an important role in the allocation process. Roughly speaking, consider a situation where for an agent a_i and a set S of items we have the following properties:

$$V_i(S) \geq \alpha\mathsf{MMS}_i$$

and

$$\forall a_j \neq a_i \qquad \mathsf{MMS}_j^{n-1}(\mathcal{M} \setminus S) \geq \mathsf{MMS}_j,$$

where $V_i(S)$ is the valuation of agent a_i for subset S of items. Intuitively, since the maxmin shares of all agents except a_i for the all items other than set S are at least as much as their current maxmin shares, allocating set S to a_i cannot hurt the guarantee. In other words, given that an α-MMS allocation is possible for all agents except a_i with items not in S, we can allocate set S to agent a_i and recursively solve the problem for the rest of the agents. Although the definition of reducibility is more general than what mentioned above, the key idea is that reducible instances of the problem can be transformed into irreducible instances. More precisely, we show that in order to prove the existence of an α-MMS allocation, it only suffices to show this for α-irreducible instances of the problem (see Observation 2.1). This makes the problem substantially simpler, since α-irreducible instances of the problem have many desirable properties. For example, in such instances, the value of every agent a_i for each item is less than $\alpha\mathsf{MMS}_i$ (see Lemma 3.1). By setting $\alpha = 1/2$, this observation along with the analysis of the bag filling algorithm, proves the existence of a 1/2-MMS allocation. It is worth to mention that a special form of reducibility, where $|S| = 1$ is used in the previouse works [2, 46].

Matching allocation: At the core of the clustering part, we use a well-structured type of matching to allocate the items to the agents. Intuitively, we cluster the agents to deal with high-value or in other words *heavy* items. In order to cluster a group of agents, we find a subset T of agents and a subset S of items, together with a matching M from S to T. We choose T, S, and M in a way that (i) every item assigned to an agent has a value of at least β to him, (ii) agents who do not receive any items have a value less than β for each of the assigned items. Such an allocation requires careful application of several properties of maximal matchings in bipartite graphs described in Section 3.2. A matching with similar structural properties is previousely used

by Procaccia and Wang [46] to allocate the bundles to the agents. In this paper, we reveal more details and precisely characterise the structure of such matchings. We use such matchings in two main steps: selecting the agents for the first and second clusters and merging the items.

Cycle-envy-freeness: Envy-freeness is itself a well-known notion for fairness in the resource allocation problems. However, this notion is perhaps more applicable to the allocation of divisible goods. In our algorithm, we use a much weaker notion of envy-freeness, namely *cycle-envy-freeness*. A cycle-envy-free allocation contains no cyclic permutation of agents, such that each agent envies the next agent in the cycle. In the clustering phase, we choose a matching M in a way that preserves cycle-envy-freeness for the clustered agents. More details about this can be found in Section 3.3.

Cycle-envy-freeness plays a key role in the second phase of the algorithm. As aforementioned, our method in the assignment phase is closely related to the bag filling procedure described above. The difference is that the efficiency of our method depends on the order of the agents who receive the items. Based on the notion of cycle-envy-freeness, we prioritize the agents and, as such, we show the allocation is fair. An analogous concept is previousely used in [43], albeit with a different application than ours.

As mentioned before, our algorithm consists of two phases: (i) clustering the agents and (ii) satisfying the agents. In the first phase, we cluster the agents into three sets namely $\mathcal{C}_1$,$\mathcal{C}_2$, and $\mathcal{C}_3$. In addition to this, for $\mathcal{C}_1$ and $\mathcal{C}_2$ we also have refinement procedures to make sure the rest of the unallocated items have a low value to the agents of these clusters. In the second phase, based on a method similar to the bag filling algorithm described above, we allocate the rest of the items to the agents. A flowchart of our algorithm is depicted in Figure 1. The main steps along with brief descriptions of each step are highlighted in the flowchart. In section 3.1, we present the ideas behind each of these steps and show how the entire algorithm leads to a proper allocation.

In Appendix A, we study the case where we only have four additive agents. Procaccia and Wang [46] showed that in this case a 3/4-MMS allocation is possible. We improve this result by giving an algorithm that finds a 4/5-MMS allocation in this restricted setting. Note that this also leads to an algorithm that finds a $4/5 - \epsilon$-MMS allocation in polynomial time. Amanatidis *et al.* [2] also show that a 7/8-MMS allocation is possible when the number of agents is equal to 3. These results indicate that better bounds can be achieved for the additive setting. We believe our framework can be used as a building block to obtain better bounds (see Section 3.1 for more details).

1.2.2 Submodular, XOS, and Subadditive Agents

Although the problem was initially proposed for additive agents, it is very well-motivated to extend the definition to other classes of set functions. For instance, it is quite natural to expect an agent prefers to receive two items of value 400, rather than receiving 1000 items of value 1. Such a constraint cannot be imposed in the additive setting. However, submodular functions which encompass k-demand valuations are strong tools for modeling these constraints. Such generalizations have been made to many similar problems, including the *Santa Claus max-min fair allocation*, *welfare maximization*, and *secretary* problems [8, 25, 26, 34]. The most common classes of set functions that have been studied before are submodular, XOS, and subadditive functions. We consider the fair allocation problem when the agents' valuations are in each of these classes. In contrast to the additive setting in which finding a constant MMS allocation is trivial, the problem becomes much more subtle even when the agents' valuations are *monotone submodular*. For instance, the bag filling algorithm does not promise any constant approximation factor for submodular agents, while it is straight-forward to show it guarantees a $(1 - \alpha)$-MMS allocation for additive agents.

We begin with submodular set functions. In Section 4, we show that the fair allocation problem

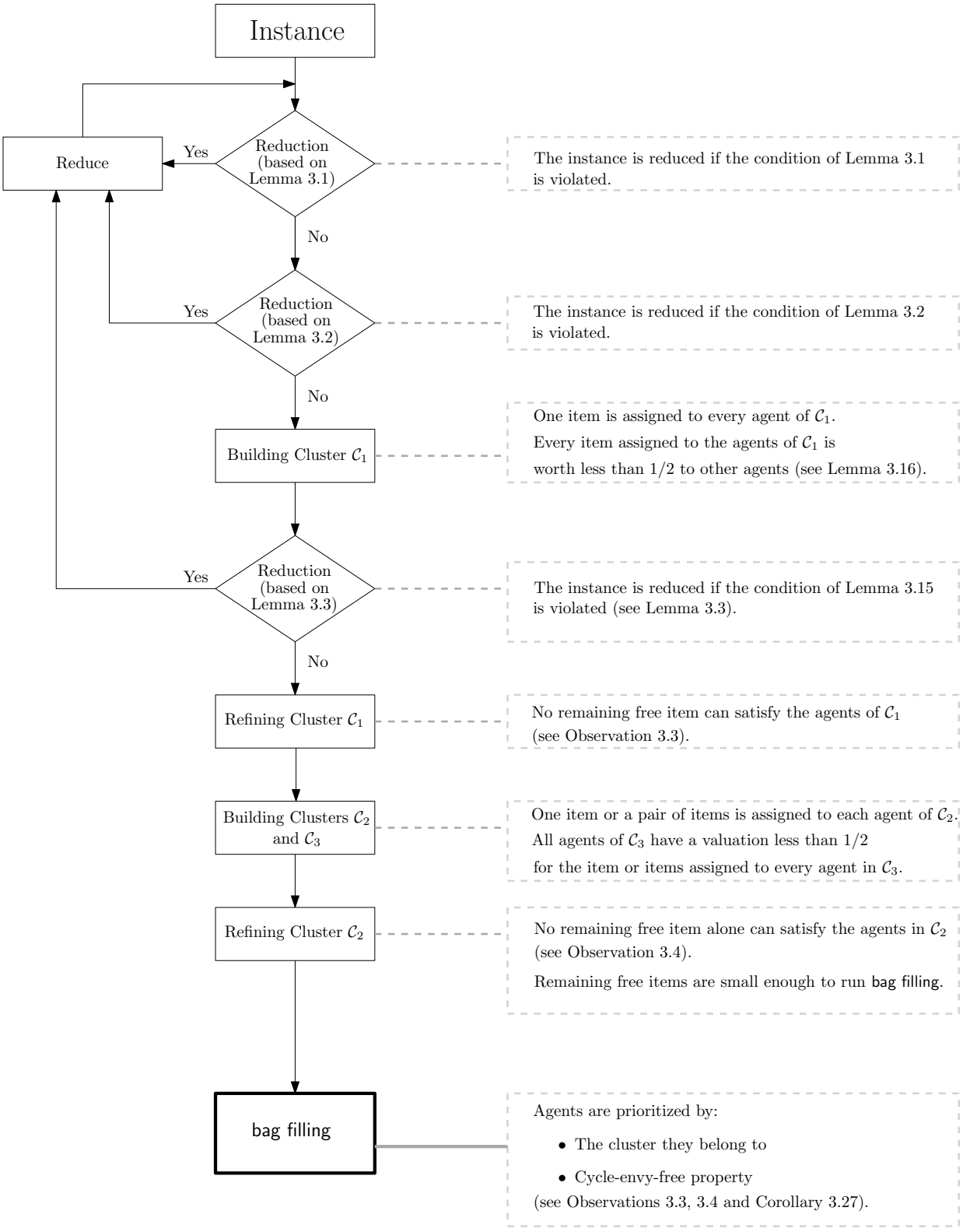

Figure 1: A flowchart of the 3/4-MMS allocation algorithm

with submodular agents admits a 1/3-MMS allocation. In addition, we show, given access to *query oracles*, one can find such an allocation in polynomial time. We further complement our result by showing that a 3/4-MMS is the best guarantee that one can hope to achieve in this setting. This is in contrast to the additive setting for which the only upper bound is that 1-MMS allocation is not always possible. We begin by stating an existential proof.

Theorem 4.7 [restated]. *The fair allocation problem with submodular agents admits a* 1/3-MMS *allocation.*

Our proof for submodular agents is fundamentally different from that of the additive setting. First, without loss of generality, we assume $\mathsf{MMS}_i = 1$ for every agent $a_i \in \mathcal{N}$. Moreover, we assume the problem is 1/3-irreducible since otherwise we can reduce the problem. Next, given a function $f(.)$, we define the *ceiling function* $f^x(.)$ as follows:

$$f^x(S) = \min\{x, f(S)\} \qquad \forall S \subseteq \mathsf{ground}(f).$$

An important property of the ceiling functions is that they preserve submodularity, fractionally subadditivity, and subadditivity (see Lemma 4.4). We define the bounded welfare of an allocation $\mathcal{A}$ as $\sum V_i^{2/3}(A_i)$. Given that, we show an allocation that maximizes the bounded welfare is 1/3-MMS. To this end, let $\mathcal{A}$ be an allocation with the maximum bounded welfare and suppose for the sake of contradiction that in such an allocation, an agent a_i receives a bundle of worth less than 1/3 to him. Since $\mathsf{MMS}_i = 1$, agent a_i can divide the items into n sets, where each set is of worth at least 1 to him. Now, we randomly select an element b_j which is *not* allocated to a_i. By the properties of submodular functions, we show the expected contribution of b_j to the valuation function of a_i is more than the expected contribution of b_j to the bounded welfare of the allocation. Therefore, there exists an item b_j such that if we allocate that item to agent a_i, the total bounded welfare of the allocation will be increased. This contradicts the maximality of the allocation.

Notice that Theorem 4.7 is only an existential proof. A natural approach to find such a solution is to start with an arbitrary allocation and iteratively increase its bounded welfare until it becomes 1/3-MMS. The main challenge though is that we do not even know what the MMS values are. Furthermore, unlike the additive setting, we do not have any PTAS algorithm that provides us a close estimate to these values. To overcome this challenge, we propose a combinatorial trick to guess these values without incurring any additional factor to our guarantee. The high level idea is to start with large numbers as estimates to the MMS values. Every time we run the algorithm on the estimated values, it either finds a desired allocation, or reports that the maxmin value of an agent is misrepresented by at least a multiplicative factor. Given this, we divide the maxmin value of that agent by that factor and continue on with the new estimates. Therefore, at every step of the algorithm, we are guaranteed that our estimates are not less than the actual MMS values. Based on this, we show that the running time of the algorithm is polynomial, and that the allocation it finds in the end has the desired properties. The reader can find a detailed discussion in Section 5.2.2.

Theorem 4.8 [restated]. *Given access to query oracles, one can find a* 1/3-MMS *allocation to submodular agents in polynomial time.*

Finally, we show that in some instances with submodular agents, no allocation is better than 3/4-MMS.

Theorem 4.1 [restated]. *For any integer number $c > 0$, there exists an instance of the fair allocation problem with $n \geq c$ submodular agents for which no allocation is better than* 3/4-MMS.

We show Theorem 4.1 by a counter-example. In this counter-example we have n agents and $2n$ items. Moreover, the valuation functions of the first $n-1$ agents are the same, but the last agent has a slightly different valuation function that makes it impossible to find an allocation which is better than 3/4-MMS. The number of agents in this example can be arbitrarily large.

In Section 5, we study the problem with fractionally subadditive (XOS) agents. We first give a 1/2 upper bound on the quality of any allocation. In other words, we show that for some instances of the problem, no allocation can guarantee anything better than 1/2-MMS when the agents valuations are XOS. This is followed by a proof to the existence of a 1/5-MMS allocation for any instance of the problem with XOS agents.

Similar to the submodular setting, we also provide an upper bound on the quality of any allocation in the XOS setting. We show Theorem 4.2 by a counter-example.

Theorem 4.2 [restated]. *For any integer number c, there is an instance of the fair allocation problem with XOS agents where $n \geq c$ and no allocation is better than* 1/2-MMS*.*

Next, we state the main theorem of this section.

Theorem 5.4 [restated]. *The fair allocation problem with XOS agents admits a* 1/5-MMS *allocation.*

Our approach for proving Theorem 5.4 is similar to the proof of Theorem 4.7. Again, we scale the valuations to make sure $\mathsf{MMS}_i = 1$ all agents and define the notion of bounded welfare, but this time as $\sum V_i^{2/5}(A_i)$. However, as XOS functions do not adhere to the nice structure of submodular functions, we use a different analysis to prove this theorem. Let $\mathcal{A}$ be an allocation with the maximum bounded welfare. In case all agents receive a value of at least 1/5, the proof is complete. Otherwise, let a_i be an agent that receives a set of items whose value to him is less than 1/5. In contrast to the submodular setting, giving no item alone to a_i can guarantee an increase in the bounded welfare of the allocation. However, this time, we show there exists a set S of items such that if we take them back from their recipients and instead allocate them to agent a_i, the bounded welfare of the allocation increases. The reason this holds is the following: since $\mathsf{MMS}_i = 1$, agent a_i can split the items into $2n$ sets where every set is worth at least 2/5 to a_i, otherwise the problem is 1/5-reducible (see Lemma 5.3). Moreover, since the valuation functions are XOS, we show that giving one of these $2n$ sets to a_i will increase the bounded welfare of the allocation. Therefore, if $\mathcal{A}$ is maximal, then $\mathcal{A}$ is also 1/5-MMS.

Finally, we show that a 1/8-MMS allocation in the XOS setting can be found in polynomial time. Our algorithm only requires access to demand and XOS oracles. Note that this bound is slightly worse than our existential proof due to some computational hardnesses. However, the blueprint of the algorithm is based on the proof of Theorem 5.4.

Theorem 5.5 [restated]. *Given access to demand and XOS oracles, we can find a* 1/8-MMS *allocation for the problem with XOS agents in polynomial time.*

We start with an arbitrary allocation and increase the bounded welfare until the allocation becomes 1/8-MMS. The catch is that if the allocation is not 1/8-MMS, then there exists an agent a_i and a set S of items such that if we take back these items from their current recipients and allocate them to agent a_i, the bounded welfare of the allocation increases. In order to increase the bounded welfare, there are two computational barriers that need to be lifted. First, similar to the submodular setting, we do not have any estimates to the MMS values. Analogously, we resolve the

first issue by iteratively guessing the MMS values. The second issue is that in every step of the algorithm, we have to find a set S of items to allocate to an agent a_i that results in an increase in the bounded welfare. Such a set S cannot be trivially found in polynomial time. That is where the demand and XOS oracles take part. In Section 5.2.1 we show how to find such a set in polynomial time. The high-level idea is the following: first, by accessing the XOS oracles, we determine the contribution of every item to the bounded welfare of the allocation. Next, we set the price of every element equal to three times the contribution of that element to the bounded welfare and run the demand oracle to find which subset has the highest profit for agent a_i. We show this subset has a value of at least $1/4$ to a_i. Next, we sort the elements of this set based on the ratio of contribution to the overall value of the set over the price of the item, and select a prefix for them that has a value of at least $1/4$ to a_i. Finally, we argue that allocating this set to a_i increases the bounded welfare of the allocation by at least some known lower bound. This, married with the combinatorial trick to guess the MMS values, gives us a polynomial time algorithm to find a $1/8$-MMS allocation.

Note that an immediate corollary of Theorems 5.5 and 4.8 is a polynomial time algorithm for approximating the maxmin value of a submodular and an XOS function within factors $1/3$ and $1/8$, respectively.

Corollary 1.1 *Let f be a submodular/XOS function on a set of ground elements S, and let n be an integer number. Given access to query oracle/demand and XOS oracles of f, we can partition the elements of S into n disjoint subsets $S_1, S_2, \ldots, S_n$ such that*

$$\min_{i=1}^{n} f(S_i) \geq c \cdot \mathsf{MMS}_f^n$$

where MMS_f^n denotes the maxmin value for function f on n subsets. Constant c equals $1/3$ if f is submodular and is equal to $1/8$ for the XOS case.

Finally, we investigate the problem when the agents are subadditive and present an existential proof based on a reduction to the XOS setting. In Section 6, we present a lemma that enables us to reduce the problem with subadditive agents to the case where agents are XOS.

Lemma 1.2 *Given a subadditive set function $f(.)$ which is defined on a set $\mathsf{ground}(f)$ and an integer number n, there exists an XOS function $g(.)$ such that*

$$\mathsf{MMS}_g^n \geq \mathsf{MMS}_f^n / \left(2\lceil \log |\mathsf{ground}(f)| \rceil\right)$$

and $g(S) \leq f(S)$ for every set $S \subseteq \mathsf{ground}(f)$.

Proof of Lemma 1.2 follows from the known techniques for reducing subadditive valuations to XOS. For the sake of completeness, we bring a formal proof in Section 6.

Theorem 6.3 [restated]. *The fair allocation problem with subadditive agents admits a $1/10\lceil \log m \rceil$-MMS allocation.*

2 Preliminaries

Throughout this paper we assume the set of agents is denoted by $\mathcal{N}$ and the set of items is referred to by $\mathcal{M}$. Let $|\mathcal{N}| = n$ and $|\mathcal{M}| = m$, we refer to the agents by a_i and to the items by b_i, i.e., $\mathcal{N} = \{a_1, a_2, \ldots, a_n\}$ and $\mathcal{M} = \{b_1, b_2, \ldots, b_m\}$. We denote the valuation of agent a_i for a set S of

items by $V_i(S)$. Our interest is in valuation functions that are monotone and non-negative. More precisely, we assume $V_i(S) \geq 0$ for every agent a_i and set $S \subseteq \mathcal{M}$, and for every two sets S_1 and S_2 we have

$$\forall a_i \in \mathcal{N} \qquad\qquad V_i(S_1 \cup S_2) \geq \max\{V_i(S_1), V_i(S_2)\}.$$

Due to obvious impossibility results for the general valuation functions[4], we restrict our attention to four classes of set functions:

- **Additive**: A set function $V(.)$ is additive if $V(S_1) + V(S_2) = V(S_1 \cup S_2) - V(S_1 \cap S_2)$ for every two sets $S_1, S_2 \in \mathsf{ground}(V)$.
- **Submodular**: A set function $V(.)$ is submodular if $V(S_1) + V(S_2) \geq V(S_1 \cup S_2) - V(S_1 \cap S_2)$ for every two sets $S_1, S_2 \in \mathsf{ground}(V)$.
- **Fractionally Subadditive (XOS)**: An XOS set function $V(.)$ can be shown via a finite set of additive functions $\{V_1, V_2, \ldots, V_\alpha\}$ where $V(S) = \max_{i=1}^{\alpha} V_i(S)$ for any set $S \subseteq \mathsf{ground}(V)$.
- **Subadditive**: A set function $V(.)$ is subadditive if $V(S_1) + V(S_2) \geq V(S_1 \cup S_2)$ for every two sets $S_1, S_2 \subseteq \mathsf{ground}(V)$.

For additive functions, we assume the value of the function for every element is given in the input. However, representing other classes of set functions requires access to oracles. For submodular functions, we assume we have access to *query oracle* defined below. Query oracles are great identifier for submodular functions, however, they are too weak when it comes to XOS and subadditive settings. For such functions, we use a stronger oracle which is called *demand oracle*. It is shown that for some functions, such as gross substitutes, a demand oracle can be implemented via a query oracle in polynomial time [42]. In addition to this, we consider a special oracle for XOS functions which is called *XOS oracle*. Access to query oracles for submodular functions, XOS oracle for XOS functions, and demand oracles for XOS and subadditive functions are quite common and have been very fruitful in the literature [19, 25, 26, 27, 29, 42, 50]. In what follows, we formally define the oracles:

- **Query oracle:** Given a function f, a query oracle $\mathcal{O}$ is an algorithm that receives a set S as input and computes $f(S)$ in time $O(1)$.
- **Demand oracle:** Given a function f, a demand oracle $\mathcal{O}$ is an algorithm that receives a sequence of prices $p_1, p_2, \ldots, p_n$ as input and finds a set S such that

  $$f(S) - \sum_{e \in S} p_e$$

  is maximized. We assume the running time of the algorithm is $O(1)$.
- **XOS oracle:** (defined only for an XOS functions f) Given a set S of items, it returns the additive representation of the function that is maximized for S. In other words, it reveals the contribution of each item in S to the value of $f(S)$.

[4] If the valuation functions are not restricted, no approximation guarantee can be achieved. For instance consider the case where we have two agents and 4 items. Agent a_1 has value 1 for sets $\{b_1, b_2\}$ and $\{b_3, b_4\}$ and 0 for the rest of the sets. Similarly, agent a_2 has value 1 for sets $\{b_1, b_3\}$ and $\{b_2, b_4\}$ and 0 for the rest of the sets. In this case, no allocation can provide both of the agents with sets which are of non-zero value to them.

Let Π_r be the set of all partitionings of $\mathcal{M}$ into r disjoint subsets. For every r-partitioning $P^* \in \Pi_r$, we denote the partitions by $P_1^*, P_2^*, \ldots, P_r^*$. For a set function $f(.)$, we define $\mathsf{MMS}_f^r(\mathcal{M})$ as follows:

$$\mathsf{MMS}_f^r(\mathcal{M}) = \max_{P^* \in \Pi_r} \min_{1 \leq j \leq r} f(P_j^*).$$

For brevity we refer to $\mathsf{MMS}_{f_i}^n(\mathcal{M})$ by MMS_i.

An allocation of items to the agents is a vector $\mathcal{A} = \langle A_1, A_2, \ldots, A_n \rangle$ where $\bigcup A_i = \mathcal{M}$ and $A_i \cap A_j = \emptyset$ for every two agents $a_i, a_j \in \mathcal{N}$. An allocation $\mathcal{A}$ is α-MMS, if every agent a_i receives a subset of the items whose value to that agent is at least α times MMS_i. More precisely, $\mathcal{A}$ is α-MMS if and only if $V_i(A_i) \geq \alpha \mathsf{MMS}_i$ for every agent $a_i \in \mathcal{N}$.

We define the notion of *reducibility* for an instance of the problem as follows.

Definition 2.1 *We say an instance of the problem is α-reducible, if there exist a set $T \subset \mathcal{N}$ of agents, a set S of items, and an allocation $\mathcal{A} = \langle A_1, A_2, \ldots, A_n \rangle$ of S to agents of T such that*

$$\forall a_i \in T \qquad\qquad V_i(A_i) \geq \alpha \mathsf{MMS}_i$$

and

$$\forall a_i \notin T \qquad \mathsf{MMS}_{V_i}^{n-|T|}(\mathcal{M} \setminus S) \geq \mathsf{MMS}_i.$$

Similarly, we call an instance *α-irreducible* if it is not α-reducible. The intuition behind Definition 2.1 is the following: In order to prove the existence of an α-MMS allocation for every instance of the problem, it only suffices to prove this for the α-irreducible instances.

Observation 2.1 *Every instance of the fair allocation problem admits an α-MMS allocation if this holds for all α-irreducible instances.*

Proof. Suppose for the sake of contradiction that all α-irreducible instances of the problem admit an α-MMS allocation, but there exists an α-reducible instance of the problem which does not admit any α-MMS allocation. Among all such instances, we consider the one with the lowest number of agents. Since this instance is α-reducible, there exists a subset T of agents and a subset S of items such that an allocation of S to agents of T provides each of them with a valuation of at least $\alpha \mathsf{MMS}_i$. Moreover, the rest of the items and agents make an instance of the problem with a smaller n. Thus, an α-MMS allocation can satisfy the rest of the agents and hence the instance admits an α-MMS allocation. This contradicts the assumption. □

The reducibility argument plays an important role in both the existential proofs and algorithms that we present in the paper. As we see in Sections 3.2 and 5, irreducible instances of the problem exhibit several desirable properties for additive and non-additive agents. We take advantage of these properties to improve the approximation guarantee of the problem for different classes of set functions.

3 Additive Agents[5]

In this section we study the fair allocation problem in the additive setting. We present a proof to the existence of a 3/4-MMS allocation when the agents are additive. This improves upon the work of Procaccia and Wang [46] wherein the authors prove a 2/3-MMS allocation exists for any

[5]We have created a website at https://www.cs.umd.edu/∼saeedrez/fair.html for the implemented algorithm and all related materials.

combination of additive agents. As we show, our proof is constructive; given an algorithm that determines the MMS of an additive set function within a factor α, we can implement an algorithm that finds a $3/(4\alpha)$-MMS allocation in polynomial time. This married with the PTAS algorithm of Epstein and Levin [22] for finding the MMS values, results is an algorithm that finds a $3/(4+\epsilon)$-MMS allocation in polynomial time.

The main idea behind the 3/4-MMS allocation is *clustering* the agents. Roughly speaking, we categorize the agents into three clusters, namely $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$. We show that the valuation functions of the agents within each cluster show similar behaviors. Along the clustering process, we allocate the heavy items (the items that have a valuation of at least 1/4 to some agents) to the agents. By Observation 2.1, proving a 3/4-MMS guarantee can be narrowed down to only 3/4-irreducible instances. The 3/4-irreducibility of the problem guarantees that after the clustering process, the remaining items are light. This enables us to run a bag filling process to satisfy the agents. In order to prove the correctness of the algorithm, we take advantage of the properties of each cluster separately.

The organization of this section is summarized in the following: we start by a brief and abstract explanation of the ideas in Section 3.1. In Section 3.2 we study the properties of the additive setting and state the main observations that later imply the correctness of our algorithm. Next, in Section 3.3 we discuss a method for clustering the agents and in Section 3.4 we show how we allocate the items to the agents of each cluster to ensure a 3/4-MMS guarantee. Finally, in Section 3.6 we explain the implementation details and prove a polynomial running time for the proposed algorithm.

Throughout this section, we assume $\mathsf{MMS}_i = 1$ for all agents $a_i \in \mathcal{N}$. This is without loss of generality for the existential proof since one can scale the valuation functions to impose this constraint. However, the computational complexity of the allocation will be affected by this assumption since determining the MMS of an additive function is NP-hard [22]. That said, we show in Section 3.6 that this challenge can be overcome by incurring an additional $1 + \epsilon$ factor to the approximation guarantee.

For brevity, we defer the proofs of Sections 3.2, 3.3, 3.4, and 3.5 to Appendices B,C,D, and E, respectively.

3.1 A Brief Overview of the Algorithm

The purpose of this section is to present an abstract overview over the ideas behind our algorithm for finding a 3/4-MMS allocation in the additive setting. For simplicity, we start with a simple 1/2-MMS algorithm mentioned in Section 1.2. Recall that the bag filling procedure guarantees a $1 - \alpha$ approximation solution when the valuations of the agents for each item is smaller than α. Furthermore, we know that in every α-irreducible instance, all the agents have a value less than α for every items. Thus, the following simple procedure yields a 1/2-MMS allocation:

(i). Reduce the problem until no agent has a value more than 1/2 for any item.

(ii). Allocate the items to the agents via a bag filling procedure.

Figure 2 shows a flowchart for this algorithm.

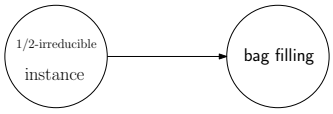

Figure 2: 1/2-MMS Algorithm

We can extend the idea in 1/2-MMS algorithm to obtain a more efficient algorithm. Here is the sketch of the 2/3-MMS algorithm: consider a 2/3-irreducible instance of the problem. In this instance, we have no item with a value more than or equal to 2/3 to any agent. Nevertheless, the items are not yet small enough to run a bag filling procedure. The idea here is to divide the agents into two clusters $\mathcal{C}_1$ and $\mathcal{C}_2$. Along this clustering, the items with a value in range $[1/3, 2/3)$ are given to the agents. In particular, one item is allocated to every agent in $\mathcal{C}_1$ that is worth at least 1/3 to him. Next, we refine Cluster $\mathcal{C}_1$. In the refining procedure, if any remaining item could singly satisfy an agent in $\mathcal{C}_1$, we do so. After building $\mathcal{C}_1$ and $\mathcal{C}_2$ and refining $\mathcal{C}_1$, the remaining items preserve the following two invariants:

(i). Value of every remaining item is less than 1/3 to every remaining agent.

(ii). No remaining item can singly satisfy an agent in $\mathcal{C}_1$ (regarding the item that is already allocated to them)

These two invariants enable us to run a bag filling procedure over the remaining items. For this case, the bag filling procedure must be more intelligent: in the case that multiple agents are qualified to receive the items of the bag, we prioritize the agents. Roughly speaking, the priorities are determined by two factors: the cluster they belong to, and the cycle-envy-freeness property of the agents in $\mathcal{C}_1$. In Figure 3 you can see a flowchart for this algorithm.

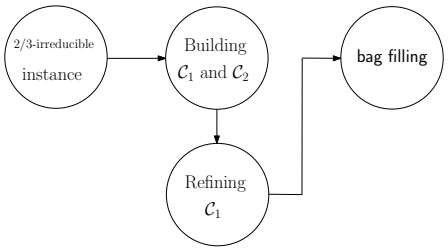

Figure 3: 2/3-MMS Algorithm

Our method for a 3/4-MMS allocation takes one step further from the previous 2/3-MMS algorithm. Again, we assume that the input is 3/4-Irreducible since otherwise it can be further simplified. Via similar ideas, we build Cluster $\mathcal{C}_1$ and refine it. Next, we build Clusters $\mathcal{C}_2$ and $\mathcal{C}_3$ and refine $\mathcal{C}_2$. After refining Cluster $\mathcal{C}_2$, the following invariants are preserved for the remaining items:

(i). Almost every remaining item has a value less than 1/4 to every remaining agent. More precisely, for every remaining agent a_i, there is at most one remaining item b_j with $V_i(\{b_j\}) \geq 1/4$.

(ii). No remaining item can singly satisfy an agent in $\mathcal{C}_1$ and $\mathcal{C}_2$ (regarding the item that is already allocated to them).

Finally, we run a bag filling procedure. Again, in the bag filling procedure, the priorities of the agents are determined by the cluster they belong to, and the cycle-envy-freeness of the clusters. In Figure 4 you can see the flowchart of the algorithm.

Our assumption is that the input is 3/4-irreducible. Hence, we describe our algorithm in two phases: a clustering phase and the bag filling phase, as shown in Figure 5. In Section 3.6 we show that all the steps of the algorithm can be implemented in polynomial time. Furthermore, we show that the assumption that the input is 3/4-irreducible is without loss of generality. In fact, in Section

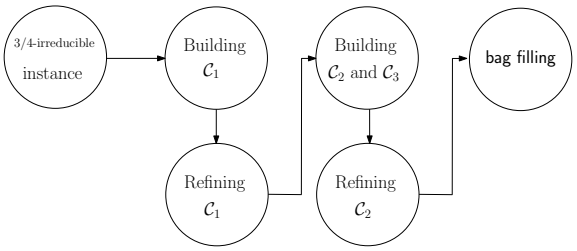

Figure 4: 3/4-MMS Algorithm

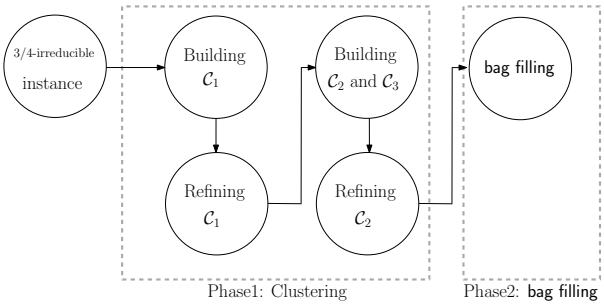

Figure 5: Algorithm Phases

3.6 we show that it suffices to check some invariants of irreducibility to be held in certain points of the algorithm. In Figure 1, these steps are specified with caption *Reduction*.

As a future work, one can consider a more generalized form of this algorithm, where the agents are divided into more than 3 clusters (see Figure 6). We believe that this generalization might yield a $(1-\epsilon)$-MMS allocation, where ϵ is a small value that depends on the number of agents. However, such a generalization is faced with two main barriers. First, In order to extend the idea to more than 3 clusters, we need a generalized form of Lemmas 3.2 and 3.3 for more than two items. Furthermore, a challenging part of our approximation proof is to show that the second cluster is empty at the end of the algorithm. For this, we define a graph on the items in the second cluster and prove some bounds on the number of edges in this graph. To extend the idea for more clusters, we need to define hypergraphs on the items in the clusters and show similar bounds, which requires deeper and more complicated techniques..

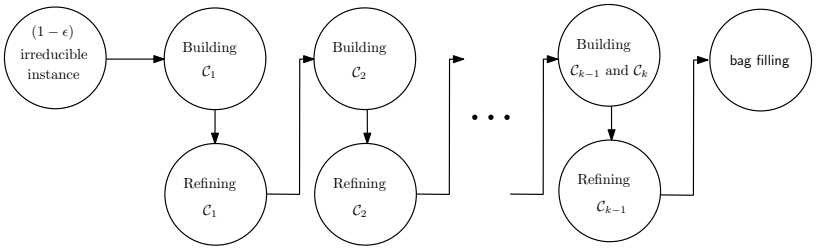

Figure 6: Generalizing the algorithm into k clusters

Before presenting the algorithm, in Section 3.2 we discuss the consequences of irreducibility and techniques to build the clusters and preserving cycle-envy-freeness in each cluster. Next, we describe the algorithm in more details.

3.2 General Definitions and Observations

Throughout this section we explore the properties of the fair allocation problem with additive agents.

3.2.1 Consequences of Irreducibility

Since the objective is to prove the existence of a 3/4-MMS allocation, by Observation 2.1, it only suffices to show every 3/4-irreducible instance of the problem admits a 3/4-MMS allocation. Therefore, in this section we provide several properties of the 3/4-irreducible instances. We say a set S of items *satisfies* an agent a_i if and only if $V_i(S) \geq 3/4$. Perhaps the most important consequence of irreducibility is a bound on the valuation of the agents for every item. In the following we show if the problem is 3/4-irreducible, then no agent has a value of 3/4 or more for an item.

Lemma 3.1 *For every α-irreducible instance of the problem we have*

$$\forall a_i \in \mathcal{N}, b_j \in \mathcal{M} \qquad V_i(b_j) < \alpha.$$

In other words, Lemma 3.1 states that in a 3/4-irreducible instance of the problem, no item alone can satisfy an agent.

It is worth mentioning that the proof for Lemma 3.1 does not rely on additivity of the valuation functions and holds as long as the valuations are monotone. Thus, regardless of the type of the valuation functions, one can assume that in any α-irreducible instance, value of any item is less than α for any agent. Hence the statement carries over to the submodular, XOS, and subadditive settings.

As a natural generalization of Lemma 3.1, we show a similar observation for every pair of items. However, this involves an additional constraint on the valuation of the other agents for the pertinent items. In contrast to Lemma 3.1, Lemmas 3.2 and 3.3 are restricted to additive setting and their results do not hold in more general settings.

Lemma 3.2 *If the problem is 3/4-irreducible and $V_i(\{b_j, b_k\}) \geq 3/4$ holds for an agent $a_i \in \mathcal{N}$ and items $b_j, b_k \in \mathcal{M}$, then there exists an agent $a_{i'} \neq a_i$ such that*

$$V_{i'}(\{b_j, b_k\}) > 1$$

According to Lemma 3.2, in every 3/4-irreducible instance of the problem, for every agent a_i and items b_j, b_k, either $V_i(\{b_j, b_k\}) < 3/4$ or there exists another agent $a_{i'} \neq a_i$, such that $V_{i'}(\{b_j, b_k\}) > 1$. Otherwise, we can reduce the problem and find a 3/4-MMS allocation recursively. More generally, let $S = \{b_{j_1}, b_{j_2}, \ldots, b_{j_{|S|}}\}$ be a set of items in $\mathcal{M}$ and $T = \{a_{i_1}, a_{i_2}, \ldots, a_{i_{|T|}}\}$ be a set of agents such that

(i) $|S| = 2|T|$

(ii) For every $a_{i_a} \in T$ we have $V_{i_a}(\{b_{j_{2a-1}}, b_{j_{2a}}\}) \geq 3/4$.

(iii) For every $a_i \notin T$ we have $V_i(\{b_{j_{2a-1}}, b_{j_{2a}}\}) \leq 1$ for every $1 \leq a \leq |T|$.

then the problem is 3/4-reducible.

Lemma 3.3 *In every 3/4-irreducible instance of the problem, for every set $T = \{a_{i_1}, a_{i_2}, \ldots, a_{i_{|T|}}\}$ of agents and set $S = \{b_{j_1}, b_{j_2}, \ldots, b_{j_{|S|}}\}$ of items at least one of the above conditions is violated.*

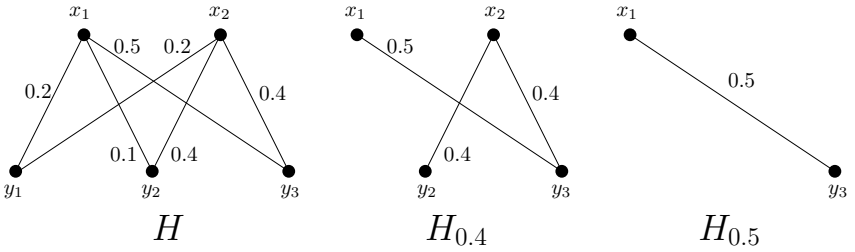

Figure 7: An example of β-filtering on a graph. After removing the edges with a value smaller than β, some vertices may become isolated. All such vertices are removed from the filtered graph.

3.2.2 Modeling the Problem with Bipartite Graphs

In our algorithm we subsequently make use of classic algorithms for bipartite graphs. Let $G = \langle V(G), E(G)\rangle$ be a graph representing the agents and the items. Moreover, let $V(G) = \mathcal{X} \cup \mathcal{Y}$ where $\mathcal{Y}$ corresponds to the agents and $\mathcal{X}$ corresponds to the items. More precisely, for every agent a_i we have a vertex $y_i \in \mathcal{Y}$ and every item b_j corresponds to a vertex $x_j \in \mathcal{X}$. For every pair of vertices $y_i \in \mathcal{Y}$ and $x_j \in \mathcal{X}$, there exists an edge $(x_j, y_i) \in E(G)$ with weight $w(x_j, y_i) = V_i(\{b_j\})$. We refer to this graph as *the value graph*.

We define an operation on the weighted graphs which we call *filtering*. Roughly speaking, a filtering is an operation that receives a weighted graph as input and removes all of the edges with weight less than a threshold from the graph. Next, we remove all of the isolated[6] vertices and report the remaining as the filtered graph. In the following we formally define the notion of filtering for weighted graphs.

Definition 3.4 *A β-filtering of a weighted graph $H\langle V(H), E(H)\rangle$, denoted by $H_\beta\langle V_\beta(H), E_\beta(H)\rangle$, is a subgraph of H where $V_\beta(H)$ is the set of all vertices in $V(H)$ incident to at least one edge of weight β or more and*

$$E_\beta(H) = \{(u, v) \in E(H) | w(u, v) \geq \beta\}.$$

For the case of the value graph, we also denote by $\mathcal{Y}_\beta$ and $\mathcal{X}_\beta$ the sets of agents and items corresponding to vertices of $V_\beta(G)$. Figure 7 illustrates an example of a graph H, together with $H_{0.4}$ and $H_{0.5}$. Note that none of the vertices in $H_{0.4}$ or $H_{0.5}$ are isolated.

Denote by a maximum matching, a matching that has the highest number of edges in a graph. In definition 3.5, we introduce our main tool for clustering the agents.

Definition 3.5 *Let $H\langle V(H), E(H)\rangle$ be a bipartite graph with $V(H) = \hat{\mathcal{X}} \cup \hat{\mathcal{Y}}$ and let M be a maximum matching of H. Define $\hat{\mathcal{Y}}_1$ as the set of the vertices in $\hat{\mathcal{Y}}$ that are not saturated by M. Also, define $\hat{\mathcal{Y}}_2$ as the set of vertices in $\hat{\mathcal{Y}}$ that are connected to $\hat{\mathcal{Y}}_1$ by an alternating path and let $\hat{\mathcal{X}}_2 = M(\hat{\mathcal{Y}}_2)$, where $M(\hat{\mathcal{Y}}_2)$ is the set of vertices in $\hat{\mathcal{X}}$ that are matched with the vertices of $\hat{\mathcal{Y}}_2$ in M. We define $F_H(M, \hat{\mathcal{X}})$ as the set of the vertices in $\hat{\mathcal{X}} \setminus \hat{\mathcal{X}}_2$.*

For a better understanding of Definition 3.5, consider Figure 8. By the definition of alternating paths, there is no edge between the saturated vertices of $F_H(M, \hat{\mathcal{X}})$ and $\hat{\mathcal{Y}}_1 \cup \hat{\mathcal{Y}}_2$. On the other hand, since M is maximum, the graph doesn't have any augmenting path. Thus, there is no edge between unsaturated vertices in $F_H(M, \hat{\mathcal{X}})$ and $\hat{\mathcal{Y}}_1 \cup \hat{\mathcal{Y}}_2$. As a result, there is no edge between $F_H(M, \hat{\mathcal{X}})$ and $\hat{\mathcal{Y}}_1 \cup \hat{\mathcal{Y}}_2$. Furthermore, $F_H(M, \hat{\mathcal{X}})$ has another important property: there exists a matching from $N(F_H(M, \hat{\mathcal{X}}))$ to $F_H(M, \hat{\mathcal{X}})$, that saturates all the vertices in $N(F_H(M, \hat{\mathcal{X}}))$, where $N(F_H(M, \hat{\mathcal{X}}))$ is the set of neighbors of $F_H(M, \hat{\mathcal{X}})$.

[6] A vertex is called isolated if no edge is incident to that vertex.

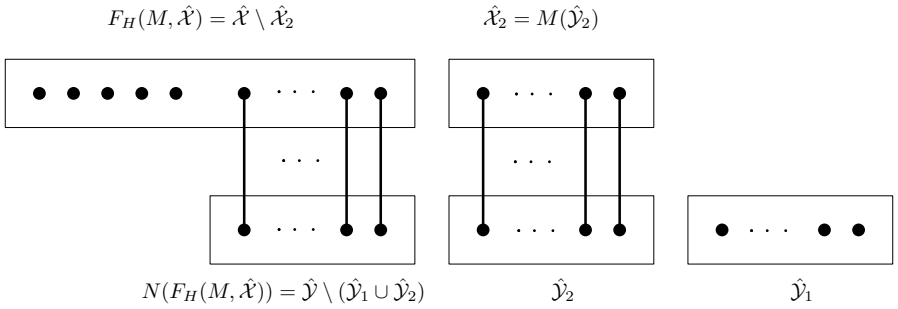

Figure 8: Definition of F_H

In Lemmas 3.7 and 3.6, we prove two remarkable properties for bipartite graphs. As a consequence of these two lemmas, Corollary 3.8 holds for every bipartite graph. We leverage the result of Corollary 3.8 in the clustering phase.

Lemma 3.6 *Let $H(V, E)$ be a bipartite graph with $V = \hat{\mathcal{X}} \cup \hat{\mathcal{Y}}$ and let M be a maximum matching of H. Then, for every set $T \subseteq \hat{\mathcal{X}} \setminus F_H(M, \hat{\mathcal{X}})$ we have $|N(T)| > |T|$, where $N(T)$ is the set of neighbors of T.*

Lemma 3.7 *For a bipartite graph $H(V, E)$ with $V = \hat{\mathcal{X}} \cup \hat{\mathcal{Y}}$, $F_H(M, \hat{\mathcal{X}}) = \emptyset$ holds, if and only if for all $T \subseteq \hat{\mathcal{X}}$ we have $|N(T)| > |T|$, where $N(T)$ is the set of neighbors of T.*

Corollary 3.8 (of Lemmas 3.7 and 3.6) *Let $H(V, E)$ be a bipartite graph with $V = \hat{\mathcal{X}} \cup \hat{\mathcal{Y}}$ and let M be a maximum matching of H. Furthermore, let $H'(V', E')$ be the induced sub-graph of H, with $V' = \hat{\mathcal{X}}' \cup \hat{\mathcal{Y}}'$, where $\hat{\mathcal{X}}' = \hat{\mathcal{X}} \setminus F_H(M, \hat{\mathcal{X}})$ and $\hat{\mathcal{Y}}' = \hat{\mathcal{Y}} \setminus N(F_H(M, \hat{\mathcal{X}}))$. Then, for any maximum matching M' of H', $F_{H'}(M', \hat{\mathcal{X}}') = \emptyset$ holds.*

3.2.3 Cycle-envy-freeness and MCMWM

In the algorithm, we satisfy each agent in two steps. More precisely, we allocate each agent two sets of items that are together of worth at least $3/4$ to him. We denote the first set of items allocated to agent a_i by f_i and the second set by g_i. Moreover, we attribute the agents with labels *satisfied*, *unsatisfied*, and *semi-satisfied* in the following way:

(i). An agent a_i is satisfied if $V_i(f_i \cup g_i) \geq 3/4$.

(ii). An agent a_i is semi-satisfied if $f_i \neq \emptyset$ but $g_i = \emptyset$. In this case we define $\epsilon_i = 3/4 - V_i(f_i)$.

(iii). An agent a_i is unsatisfied if $f_i = g_i = \emptyset$.

As we see, the algorithm maintains the property that for every semi-satisfied agent a_i, $V_i(f_i) \geq 1/2$ holds and hence, $\epsilon_i < 1/4$.

To capture the competition between different agents, we define an attribution for an ordered pair of agents. We say a semi-satisfied agent envies another semi-satisfied agent, if he prefers to switch sets with the other agent.

Definition 3.9 *Let T be a set of semi-satisfied agents. An agent $a_i \in T$ envies an agent $a_j \in T$, if $V_i(f_j) \geq V_i(f_i)$. Also, we call an agent $a_i \in T$ a winner of T, if a_i envies no other agent in T. Similarly, we call an agent a_i a loser of T, if no other agent in T envies a_i.*

Note that it could be the case that an agent a_i is both a loser and a winner of a set T of agents. Based on Definition 3.9, we next define the notion of *cycle-envy-freeness*.

Definition 3.10 *We call a set T of semi-satisfied agents cycle-envy-free, if every non-empty subset of T contains at least one winner and one loser.*

Let C be a cycle-envy-free set of semi-satisfied agents. Define the representation graph of C as a digraph $G_C(V(G_C), \overrightarrow{E}(G_C))$, such that for any agent $a_i \in C$, there is a vertex v_i in $V(G_C)$ and there is a directed edge from v_i to v_j in $\overrightarrow{E}(G_C)$, if a_i envies a_j. In Lemma 3.11, we show that G_C is acyclic.

Lemma 3.11 *For every cycle-envy-free set of semi-satisfied agents C, G_C is a DAG.*

Definition 3.12 *A topological ordering of a cycle-envy-free set C of semi-satisfied agents, is a total order $\prec_O$ corresponding to the topological ordering of the representation graph G_C. More formally, for the agents $a_i, a_j \in C$ we have $a_i \prec_O a_j$ if and only if v_i appears before v_j, in the topological ordering of G_C.*

Note that in the topological ordering of a cycle-envy-free set C of semi-satisfied agents, if $a_i \in C$ envies $a_j \in C$, then $a_i \prec_O a_j$.

Observation 3.1 *Let C be a cycle-envy-free set of semi-satisfied agents. Then, for every agent $a_i \in C$ such that $a_j \prec_O a_i$, we have:*

$$V_i(f_j) \leq 3/4 - \epsilon_i.$$

We define a maximum cardinality maximum weighted matching of a weighted graph as a matching that has the highest number of edges and among them the one that has the highest total sum of edge weights. For brevity we call such a matching an MCMWM. In Lemma 3.13, we show that an MCMWM of a weighted bipartite graph has certain properties that makes it useful for building cycle-envy-free clusters.

Lemma 3.13 *Let $H\langle V(H), E(H)\rangle$ be a weighted bipartite graph with $V(H) = \hat{\mathcal{X}} \cup \hat{\mathcal{Y}}$ and let $M = \{(\hat{x}_1, \hat{y}_1), ..., (\hat{x}_k, \hat{y}_k)\}$ be an MCMWM of H. Then, for every subset $T \subseteq \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_k\}$, the following conditions hold:*

(i). There exists a vertex $\hat{y}_j \in T$ which is a winner in T, i.e., $w(\hat{x}_j, \hat{y}_j) \geq w(\hat{x}_i, \hat{y}_j)$, for all $\hat{x}_i \in M(T)$ and $(\hat{x}_i, \hat{y}_j) \in E(H)$.

(ii). There exists a vertex $\hat{y}_j \in T$ which is a loser in T, i.e., $w(\hat{x}_i, \hat{y}_i) \geq w(\hat{x}_j, \hat{y}_i)$, for all $\hat{y}_i \in T$ and $(\hat{x}_j, \hat{y}_i) \in E(H)$.

(iii). For any vertex $\hat{y}_i \in T$ and any unsaturated vertex $\hat{x}_j \in \hat{\mathcal{X}}$ such that $(\hat{x}_j, \hat{y}_i) \in E(H)$, $w(\hat{x}_i, \hat{y}_i) \geq w(\hat{x}_j, \hat{y}_i)$.

where $M(T)$ is the set of vertices which are matched by the vertices of T in M.

Notice the similarities of the first and the second conditions of Lemma 3.13 with the conditions of the winner and loser in Definition 3.9. In Section 3.3, we assign items to the agents based on an MCMWM of the value-graph. Lemma 3.13 ensures that such an assignment results in a cycle-envy-free set of semi-satisfied agents.

3.3 Phase 1: Building the Clusters

In this section, we explain our method for clustering the agents. Intuitively, we divide the agents into three clusters $\mathcal{C}_1, \mathcal{C}_2$ and $\mathcal{C}_3$. As mentioned before, during the algorithm, two sets of items f_i, g_i are allocated to each agent a_i. Throughout this section, we prove a set of lemmas that are labeled as *value-lemma*. In these lemmas we bound the value of f_i and g_i allocateed to any agent for other agents. A summary of these lemmas is shown in Tables 3, 4 and 5.

After constructing each cluster, we refine that cluster. In the refinement phase of each cluster, we target a certain subset of the remaining items. If any item in this subset could satisfy an agent in the recently created cluster, we allocate that item to the corresponding agent. The goal of the refinement phase is to ensure that the remaining items in the targeted subset are light enough for the agents in that cluster, i.e., none of the remaining items can satisfy an agent in this cluster.

We denote by $\mathcal{S}$, the set of satisfied agents. In addition, denote by $\mathcal{S}_1, \mathcal{S}_2$, and $\mathcal{S}_3$ the subsets of $\mathcal{S}$, where $\mathcal{S}_i$ refers to the agents of $\mathcal{S}$ that previously belonged to $\mathcal{C}_i$. Furthermore, we use $\mathcal{S}_1^r$ and $\mathcal{S}_2^r$ to refer to the agents of $\mathcal{S}_1$ and $\mathcal{S}_2$ that are satisfied in the refinement phases of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively.

3.3.1 Cluster $\mathcal{C}_1$

Consider the filtering $G_{1/2}\langle V_{1/2}(G), E_{1/2}(G)\rangle$ of the value-graph G and let M be an MCMWM of $G_{1/2}$. We define Cluster $\mathcal{C}_1$ as the set of agents whose corresponding vertex is in $N(F_{G_{1/2}}(M, \mathcal{X}_{1/2}))$.

For brevity, denote by $V_{\mathcal{C}_1}$ the set of vertices in $V(G)$ that correspond to the agents of $\mathcal{C}_1$. In other words:

$$V_{\mathcal{C}_1} = N(F_{G_{1/2}}(M, \mathcal{X}_{1/2})).$$

Also, let $F_{G_{1/2}}(M, \mathcal{X}_{1/2})$ be $U_1 \cup S_1$, where U_1 is the set of unsaturated vertices in $F_{G_{1/2}}(M, \mathcal{X}_{1/2})$ and S_1 is the set of the saturated vertices. For each edge $(x_j, y_i) \in M$ such that $x_j \in S_1$, we allocate item b_j to agent a_i. More precisely, we set $f_i = \{b_j\}$. Since $w(x_j, y_i) \geq 1/2$, we have:

$$\forall a_k \in \mathcal{C}_1 \qquad V_k(f_k) \geq 1/2.$$

According to the definition of ϵ_i, we have

$$\forall a_k \in \mathcal{C}_1 \qquad \epsilon_k \leq 1/4. \tag{1}$$

By the definition of $F_{G_{1/2}}$, for every agent which is not in $\mathcal{C}_1$, the condition of Lemma 3.14 holds. Note that all the agents that are not in $\mathcal{C}_1$, belong to either $\mathcal{C}_2$ or $\mathcal{C}_3$.

Lemma 3.14 (value-lemma) *For all $a_i \in \mathcal{C}_2 \cup \mathcal{C}_3$ we have:*

$$\forall a_j \in \mathcal{C}_1 \qquad V_i(f_j) < 1/2.$$

For each vertex $y_i \in V_{\mathcal{C}_1}$, denote by N_{y_i} the set of vertices $x_j \in \mathcal{X} \setminus \mathcal{X}_{1/2}$, where $w(x_j, y_i) \geq \epsilon_i$ and let

$$W_1 = U_1 \cup \bigcup_{y_i \in V_{\mathcal{C}_1}} N_{y_i}.$$

Note that by definition, for any vertex $x_j \in U_1$ and $y_i \notin V_{\mathcal{C}_1}$, there is no edge between x_j and y_i in $G_{1/2}$ and hence $w(x_j, y_i) < 1/2$. Also, since the rest of the vertices in W_1 are from $\mathcal{X} \setminus \mathcal{X}_{1/2}$, for any vertex y_i and $x_j \in (W_1 \setminus U_1)$, $w(x_j, y_i) < 1/2$ holds. Thus, we have the following observation:

Observation 3.2 *For every item b_j with $x_j \in W_1$ and every agent a_i with $y_i \notin V_{\mathcal{C}_1}$, $V_i(\{b_j\}) < 1/2$.*

Now, define $\mathcal{X}'$ and $\mathcal{Y}'$ as follows:

$$\mathcal{X}' = \mathcal{X} \setminus (W_1 \cup S_1),$$
$$\mathcal{Y}' = \mathcal{Y} \setminus V_{\mathcal{C}_1}.$$

Let $G'\langle V(G'), E(G')\rangle$ be the induced subgraph of G on $V(G') = \mathcal{Y}' \cup \mathcal{X}'$. We use graph G' to build Cluster $\mathcal{C}_2$.

3.3.2 Cluster $\mathcal{C}_1$ Refinement

Before building Cluster $\mathcal{C}_2$, we satisfy some of the agents in $\mathcal{C}_1$ with the items corresponding to the vertices of W_1. Consider the subgraph $G_1\langle V(G_1), E(G_1)\rangle$ of G with $V(G_1) = W_1 \cup V_{\mathcal{C}_1}$. In G_1, There is an edge between $y_i \in V_{\mathcal{C}_1}$ and $x_j \in W_1$, if $V_i(\{b_j\}) \geq \epsilon_i$. Note that $G_1\langle V(G_1), E(G_1)\rangle$ is not necessarily an induced subgraph of G. We use G_1 to satisfy a set of agents in $\mathcal{C}_1$. To this end, we first show that G_1 admits a special type of matching, described in Lemma 3.15.

Lemma 3.15 *There exists a matching M_1 in G_1, that saturates all the vertices of W_1 and for any edge $(x_i, y_j) \in M_1$ and any unsaturated vertex $y_k \in N(x_i)$, a_k does not envy a_j.*

Let M_1 be a matching of G_1 with the property described in Lemma 3.15. For every edge $(y_i, x_j) \in M_1$, we allocate item b_j to agent a_i i.e., we set $g_i = \{b_j\}$. By the definition, a_i is now satisfied. Thus, we remove a_i from $\mathcal{C}_1$ and add it to $\mathcal{S}$. Note that, after refining $\mathcal{C}_1$, none of the items whose corresponding vertex is in $\mathcal{X}' \setminus \mathcal{X}'_{1/2}$ can satisfy any remaining agent in $\mathcal{C}_1$. Thus, Observation 3.3 holds.

Observation 3.3 *For every item b_j such that $x_j \in \mathcal{X}'$, either $x_j \in \mathcal{X}'_{1/2}$ or for all $a_i \in \mathcal{C}_1$, $V_i(\{b_j\}) < \epsilon_i$.*

At this point, all the agents of $\mathcal{S}$ belong to $\mathcal{S}_1^r$. Each one of these agents is satisfied with two items, i.e., for any agent $a_j \in \mathcal{S}_1^r$, $|f_j| = |g_j| = 1$. In Lemma 3.16 we give an upper bound on $V_i(g_j)$ for every agent $a_j \in \mathcal{S}_1^r$ and every agent a_i in $\mathcal{C}_2 \cup \mathcal{C}_3$.

Lemma 3.16 (value-lemma) *For every agent $a_i \in \mathcal{C}_2 \cup \mathcal{C}_3$, we have*

$$\forall a_j \in \mathcal{S}_1^r \qquad V_i(g_j) < 1/2.$$

Lemmas 3.16 and 3.14 state that for every agent $a_i \in \mathcal{C}_2 \cup \mathcal{C}_3$ and every agent $a_j \in \mathcal{S}_1^r$, $V_i(f_j)$ and $V_i(g_j)$ are upper bounded by $1/2$. This, together with the fact that $|f_j| = |g_j| = 1$, results in Lemma 3.17.

Lemma 3.17 *For all $a_i \notin \mathcal{C}_1$, we have*

$$\mathsf{MMS}_{V_i}^{|\mathcal{N}\setminus\mathcal{S}_1^r|}(\mathcal{M} \setminus \bigcup_{y_j \in \mathcal{S}_1^r} f_j \cup g_j) \geq 1.$$

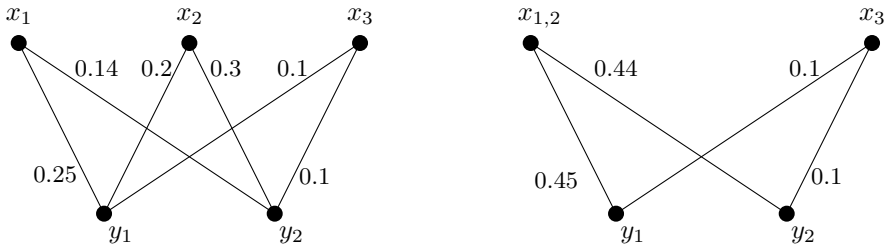

Figure 9: Merging x_1 and x_2

3.3.3 Cluster $\mathcal{C}_2$

Recall graph $G'\langle V(G'), E(G')\rangle$ as described in the last part of Section 3.3.1 and let $G'_{1/2}\langle V_{1/2}(G'), E_{1/2}(G')\rangle$ be a 1/2-filtering of G'. Lemma 3.6 states that the size of the maximum matching between $\mathcal{X}'_{1/2}$ and $\mathcal{Y}'_{1/2}$ is $|\mathcal{X}'_{1/2}|$. Also, according to Corollary 3.8, for any maximum matching M' of $G'_{1/2}$, $F_{G'_{1/2}}(M', \mathcal{X}'_{1/2})$ is empty. In what follows, we increase the size of the maximum matching in $G'_{1/2}$ by merging the vertices of $\mathcal{X}' \setminus \mathcal{X}'_{1/2}$ as described in Definition 3.18.

Definition 3.18 *For merging vertices x_i, x_j of $G'(\mathcal{X}', \mathcal{Y}')$, we create a new vertex labeled with $x_{i,j}$. Next, we add $x_{i,j}$ to $\mathcal{X}'$ and for every vertex $y_k \in \mathcal{Y}'$, we add an edge from y_k to $x_{i,j}$ with weight $w(y_k, x_i) + w(y_k, x_j)$. Finally we remove vertices x_i and x_j from $\mathcal{X}$. See Figure 9.*

In Lemmas 3.19 and 3.20, we give upper bounds on the value of the pair of items corresponding to a merged vertex. In Lemma 3.19, we show that the value of a merged vertex is less than $2\epsilon_i$ to every agent $a_i \in \mathcal{C}_1$. This fact is a consequence of Observation 3.3. Also, in Lemma 3.20, we prove that the value of the items corresponding to a merged vertex is less than 3/4 to any agent. Lemma 3.20 is a direct consequence of 3/4-irreducibility. In fact, we show that if the condition of Lemma 3.20 does not hold, then the problem can be reduced.

Lemma 3.19 *For any agent $a_k \in \mathcal{C}_1$ and any pair of vertices $x_i, x_j \in \mathcal{X}' \setminus \mathcal{X}'_{1/2}$, $V_k(\{b_i, b_j\}) < 2\epsilon_k$ holds. In particular, total value of the items that belong to a merged vertex is less than $2\epsilon_k$ for a_k.*

Lemma 3.20 *For any pair of vertices $x_i, x_j \in \mathcal{X}' \setminus \mathcal{X}'_{1/2}$ and any vertex $y_k \in \mathcal{Y}$, we have $V_k(\{b_i, b_j\}) < 3/4$.*

Corollary 3.21 (of Lemma 3.20) *For any agent a_i with $y_i \in \mathcal{Y}$, there is at most one item b_j, with $x_j \in \mathcal{X}' \setminus \mathcal{X}'_{1/2}$ and $V_i(\{b_j\}) \geq 3/8$.*

Consider the vertices in $\mathcal{X}' \setminus \mathcal{X}'_{1/2}$. We call a pair (x_i, x_j) of distinct vertices in $\mathcal{X}' \setminus \mathcal{X}'_{1/2}$ *desirable* for $y_k \in \mathcal{Y}'$, if $w(y_k, x_i) + w(y_k, x_j) \geq 1/2$. With this in mind, consider the process described in Algorithm 1.

In each step of this process, we find an MCMWM M' of $G'_{1/2}$. Note that M' changes after each step of the algorithm. Next, we find a pair (x_i, x_j) of the vertices in $\mathcal{X}' \setminus \mathcal{X}'_{1/2}$ that is desirable for at least one agent in $T = \mathcal{Y}' \setminus N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))$. If no such pair exists, we terminate the algorithm. Otherwise, we select an arbitrary desirable pair (x_i, x_j) and merge them to obtain a vertex $x_{i,j}$. According to the definition of T in Algorithm 1, merging a pair (x_i, x_j) results in an augmenting path in $G'_{1/2}$. Hence, the size of the maximum matching in $G'_{1/2}$ is increased by one. Note that after the termination of Algorithm 1, either $T = \emptyset$ or no pair of vertices in $\mathcal{X}' \setminus \mathcal{X}'_{1/2}$ is desirable for any vertex in T.

Algorithm 1: Merging vertices in G'

Data: $G'(V(G'), E(G'))$

while *True* **do**
  $M' =$ MCMWM of $G'_{1/2}$;
  Find $F_{G'_{1/2}}(M', \mathcal{X}'_{1/2})$;
  $T = \mathcal{Y}' \setminus N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))$;
  $Q =$ Set of all desirable pairs in $\mathcal{X}' \setminus \mathcal{X}'_{1/2}$ for the agents in T;
  if $Q = \emptyset$ **then**
    STOP;
  else
    Select an arbitrary pair x_i, x_j from Q;
    Merge(x_i, x_j);

Lemma 3.22 *After running Algorithm 1, we have*

$$|F_{G'_{1/2}}(M', \mathcal{X}'_{1/2})| = |N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))|.$$

We define Cluster $\mathcal{C}_2$ as the set of agents that correspond to the vertices of $N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))$. Also, denote by $V_{\mathcal{C}_2}$ the vertices in $N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))$. For each agent $a_i \in \mathcal{C}_2$, we allocate the item corresponding to $M'(y_i)$ (or pair of items in case $M'(y_i)$ is a merged vertex) to a_i.

Note that we put the rest of the agents in Cluster $\mathcal{C}_3$. Therefore, Lemma 3.23 holds for all the agents of $\mathcal{C}_3$.

Lemma 3.23 (value-lemma) *For all $a_i \in \mathcal{C}_3$ we have*

$$\forall a_j \in \mathcal{C}_2, V_i(f_j) < 1/2.$$

3.3.4 Cluster $\mathcal{C}_2$ Refinement

The refinement phase of $\mathcal{C}_2$, is semantically similar to the refinement phase of $\mathcal{C}_1$. In the refinement phase of $\mathcal{C}_2$, we satisfy some of the agents of $\mathcal{C}_2$ by the items with vertices in $\mathcal{X}' \setminus \mathcal{X}'_{1/2}$. Note that none of the vertices in $\mathcal{X}' \setminus \mathcal{X}'_{1/2}$ is a merged vertex.

The refinement phase of $\mathcal{C}_2$ is presented in Algorithm 2. Let $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ be the topological ordering of the agents in $\mathcal{C}_2$ as described in Section 3.2.3 . In Algorithm 2, We start with y_{i_1} and $W_2 = \emptyset$ and check whether there exists a vertex $x_j \in \mathcal{X}' \setminus (\mathcal{X}'_{1/2} \cup W_2)$ such that $V_{i_1}(\{b_j\}) \geq \epsilon_{i_1}$. If so, we add x_j to W_2 and satisfy a_{i_1} by allocating b_j to a_{i_1}. Next, we repeat the same process for y_{i_2} and continue on to y_{i_k}. Note that at the end of the process, W_2 refers to the vertices whose corresponding items are allocated to the agents that are satisfied during the refinement step of $\mathcal{C}_2$. For convenience, let $S_2 = F_{G'_{1/2}}(M', \mathcal{X}'_{1/2})$ and define $\mathcal{X}''$ and $\mathcal{Y}''$ as follows:

$$\mathcal{X}'' = \mathcal{X}' \setminus (W_2 \cup S_2),$$

$$\mathcal{Y}'' = \mathcal{Y}' \setminus V_{\mathcal{C}_2}.$$

Let $G''\langle V(G''), E(G'')\rangle$ be the induced subgraph of G' on $V(G'') = \mathcal{X}'' \cup \mathcal{Y}''$. We use G'' to build Cluster $\mathcal{C}_3$.

Algorithm 2: Refinement of $\mathcal{C}_2$

Data: $G'(V(G'), E(G'))$
Data: $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ = Topological ordering of agents in $\mathcal{C}_2$
for $l : 1 \to k$ **do**
  if $\exists x_j \in \mathcal{X}' \setminus (\mathcal{X}'_{1/2} \cup W_2)$ *s.t.* $V_{i_1}(\{b_j\}) \geq \epsilon_{i_l})$ **then**
    $g_{i_l} = b_j$;
    $W_2 = W_2 \cup x_j$;
    $\mathcal{C}_2 = \mathcal{C}_2 \setminus a_{i_l}$;
    $\mathcal{S} = \mathcal{S} \cup a_{i_l}$;

Observation 3.4 *After running Algorithm 2, For every item b_j with $x_j \in \mathcal{X}'' \setminus \mathcal{X}''_{1/2}$ and every agent $a_i \in \mathcal{C}_2$, we have $V_i(\{b_j\}) < \epsilon_i$.*

In the following two lemmas, we give upper bounds on the value of g_i for every agent $a_i \in \mathcal{S}_2^r$. First, in Lemma 3.24, we show that for every agent $a_j \in \mathcal{C}_1$, $V_j(g_i)$ is upper bounded by ϵ_j. Furthermore, by the fact that the agents that are not selected for Clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ belong to Cluster $\mathcal{C}_3$, we show that $V_j(g_i)$ is upper bounded by $1/2$ for every agent $a_j \in \mathcal{C}_3$.

Lemma 3.24 (value-lemma) *Let $a_i \in \mathcal{S}_2^r$ be an agent that is satisfied in the refinement phase of Cluster $\mathcal{C}_2$ and a_j be an agent in $\mathcal{C}_1$. Then, $V_j(g_i) < \epsilon_j$.*

Lemma 3.25 (value-lemma) *Let $a_i \in \mathcal{S}_2^r$ be an agent that is satisfied in the refinement phase of Cluster $\mathcal{C}_2$ and a_j be an agent in $\mathcal{C}_3$. Then, $V_j(g_i) < 1/2$.*

3.3.5 Cluster $\mathcal{C}_3$.

Finally, Cluster $\mathcal{C}_3$ is defined as the set of agents corresponding to the vertices of $\mathcal{Y}''$. Let M'' be an MCMWM of $G''_{1/2}$. Note that by Lemma 3.6, all the vertices in $\mathcal{X}''_{1/2}$ are saturated by M''. For each vertex y_i that is saturated by M'', we allocate the item (or pair of items in a case that $M''(y_i)$ is a merged vertex) corresponding to $M''(y_i)$ to a_i. Unlike the previous clusters, this allocation is temporary. A semi-satisfied agent a_i in $\mathcal{C}_3$ may *lend* his f_i to the other agents of $\mathcal{C}_3$. Therefore, we have three type of agents in $\mathcal{C}_3$:

(i). **The semi-satisfied agents**: we denote the set of semi-satisfied agents in $\mathcal{C}_3$ by $\mathcal{C}_3^s$

(ii). **The borrower agents**: the agents that may borrow from a semi-satisfied agent. An agent a_j in $\mathcal{C}_3$ is a borrower, if $a_j \notin \mathcal{C}_3^s$ and $\max_{a_i \in \mathcal{C}_3^S} V_j(f_i) \geq 1/2$. We denote the set of borrower agents in $\mathcal{C}_3$ by $\mathcal{C}_3^b$.

(iii). **The free agents**: the remaining agents in $\mathcal{C}_3$. We denote the set of free agents by $\mathcal{C}_3^f$.

So far, the agents corresponding to unsaturated vertices in $\mathcal{Y}''_{1/2}$ belong to $\mathcal{C}_3^b$ and the agents corresponding to the vertices in $\mathcal{Y}'' \setminus \mathcal{Y}''_{1/2}$ are in $\mathcal{C}_3^f$. As we see, during the second phase, agents in $\mathcal{C}_3$ may change their type. For example, an agent in $\mathcal{C}_3^s$ may move to $\mathcal{C}_3^f$ or vice versa. For convenience, for every agent $a_i \in \mathcal{C}_3^b$, we define ϵ_i as follows:

$$3/4 - \max_{a_j \in \mathcal{C}_3^s} V_i(f_j) \tag{2}$$

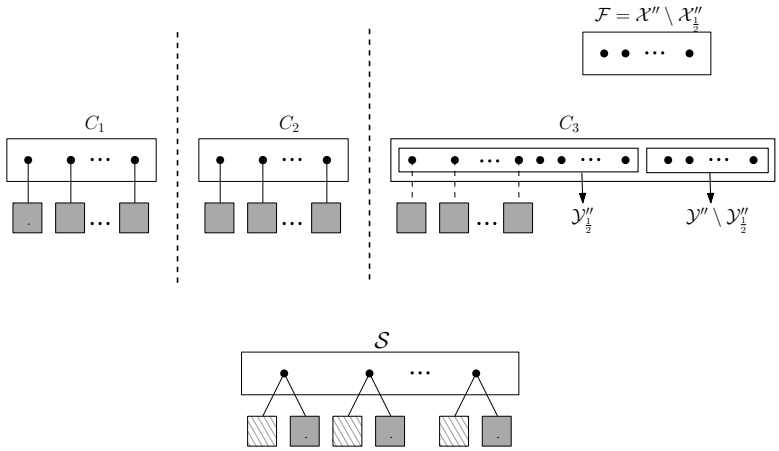

Figure 10: Overview on the state of the algorithm

Note that by the definition, $\epsilon_i \leq 1/4$ holds for every agent of $\mathcal{C}_3^b$.

In Lemma 3.26, we show that the remaining items are not *heavy* for the agents in $\mathcal{C}_3$. The main reason that Lemma 3.26 holds, is the fact that no pair of vertices is desirable for any agents in $\mathcal{C}_3$ at the end of Algorithm 1.

Lemma 3.26 *For all $a_i \in \mathcal{C}_3$ and $x_j, x_k \in \mathcal{X}'' \setminus \mathcal{X}''_{1/2}$, we have $V_i(\{b_j, b_k\}) < 1/2$.*

Corollary 3.27 (of Lemma 3.26) *For any agent $a_i \in \mathcal{C}_3$, there is at most one vertex $x_j \in \mathcal{X}'' \setminus \mathcal{X}''_{1/2}$, such that $V_i(\{b_j\}) \geq 1/4$.*

3.4 Phase 2: Satisfying the Agents

3.4.1 An Overview on the State of the Algorithm

Before going through the second phase, we present an overview of the current state of the agents and items. In Figure 10, for every agent $a_i \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{S}$, f_i is shown by a gray rectangle and for every agent $a_i \in \mathcal{S}$, g_i is shown by a hatched rectangle.

Currently, we know that every agent in $\mathcal{S}$ belongs to $\mathcal{S}_1^r$ or $\mathcal{S}_2^r$. These agents are satisfied in the refinement phases of $\mathcal{C}_1$ and $\mathcal{C}_2$. The rest of the agents will be satisfied in the second phase. For brevity, for $i \leq 2$ we use $\mathcal{S}_i^s$ to refer to the agents in $\mathcal{S}_i$ that are satisfied in the second phase. More formally,

$$\text{for } i = 1, 2 \qquad \mathcal{S}_i^s = \mathcal{S}_i \setminus \mathcal{S}_i^r.$$

Since we didn't refine Cluster $\mathcal{C}_3$, all the agents in the Cluster $\mathcal{C}_3$ are satisfied in the second phase. As mentioned in the previous section, the item allocation to the semi-satisfied agents in $\mathcal{C}_3$ is temporary; That is, we may alter such allocations later. Therefore, in Figure 10 we illustrate such allocations by dashed lines.

In this section, we denote the set of free items (the items corresponding to the vertices in $\mathcal{X}'' \setminus \mathcal{X}''_{1/2}$ at the end of the first phase) by $\mathcal{F}$. By Observations 3.3, 3.4 and Corollary 3.27, we know that the items in $\mathcal{F}$ have the following properties:

(i). For every agent a_i in $\mathcal{C}_1$, $V_i(\{b_j\}) < \epsilon_i$ holds for all $b_j \in \mathcal{F}$ (Observation 3.3).

(ii). For every agent a_i in $\mathcal{C}_2$, $V_i(\{b_j\}) < \epsilon_i$ holds for all $b_j \in \mathcal{F}$ (Observation 3.4).

Table 3: Summary of value lemmas for f_i

	$\forall a_i \in \mathcal{C}_1$	$\forall a_i \in \mathcal{C}_2$	$\forall a_i \in \mathcal{C}_3$
$\forall a_j \in \mathcal{C}_1$	-	$V_i(f_j) < 1/2$ ($\star$)	$V_i(f_j) < 1/2$ ($\star$)
$\forall a_j \in \mathcal{C}_2$	$V_i(f_j) < 3/4$ ($\ddagger$)	-	$V_i(f_j) < 1/2$ ($\dagger$)
$\forall a_j \in \mathcal{C}_3^s$	$V_i(f_j) < 3/4$($\ddagger$)	$V_i(f_j) < 3/4$($\ddagger$)	-

$\star$: Lemma 3.14 $\dagger$: Lemma 3.23 $\ddagger$: Lemma 3.28

Table 4: Summary of value lemmas for the agents in $\mathcal{S}_i^r$

	$\forall a_i \in \mathcal{C}_1$	$\forall a_i \in \mathcal{C}_2$	$\forall a_i \in \mathcal{C}_3$
$\forall a_j \in \mathcal{S}_1^r$	-	$V_i(g_j) < 1/2$ ($\star$)	$V_i(g_j) < 1/2$ ($\star$)
$\forall a_j \in \mathcal{S}_2^r$	$V_i(g_j) < \epsilon_i$($\dagger$)	-	$V_i(g_j) < 1/2$ ($\ddagger$)

$\star$: Lemma 3.16 $\dagger$: Lemma 3.24 $\ddagger$: Lemma 3.25

(iii). For every agent a_i in $\mathcal{C}_3$, there is at most one item $b_j \in \mathcal{F}$, such that $V_i(\{b_j\}) \geq 1/4$ (Corollary 3.27).

In summary, items of $\mathcal{F}$ are small enough, therefore we can run a process similar to the bag filling algorithm described earlier to allocate them to the agents. Recall that our clustering and refinement methods preserve the conditions stated in Lemmas 3.14, 3.16, 3.23, 3.24 and 3.25. In addition to this, we state Lemma 3.28 as follows.

Lemma 3.28 (value-lemma) *For every agent $a_i \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3^s$, we have*

$$\forall a_j \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \qquad V_j(f_i) < 3/4.$$

A brief summary of Lemmas 3.14, 3.16, 3.23, 3.24, 3.25 and 3.28 is illustrated in Tables 3 and 4. Moreover, since sets $\mathcal{C}_1, \mathcal{C}_2$ and $\mathcal{C}_3^s$ are cycle-envy-free, Observation 3.1 holds for these sets.

3.4.2 Second Phase: bag filling

We begin this section with some definitions. In the following, we define the notion of feasible subsets and, based on that, we define $\phi(S)$ for a feasible subset S of items.

Definition 3.29 *A subset S of items in $\mathcal{F}$ is feasible, if at least one of the following conditions are met:*

(i). There exists an agent $a_i \in \mathcal{C}_3^f$ such that $V_i(\{S\}) \geq 1/2$.

(ii). There exists an agent $a_i \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3^s \cup \mathcal{C}_3^b$ such that $V_i(\{S\}) \geq \epsilon_i$.

Definition 3.30 *For a feasible set S, we define $\Phi(S)$ as the set of agents, that set S is feasible for them.*

Recall the notion of cycle-envy-freeness and the topological ordering of the agents in a cycle-envy-free set of semi-satisfied agents. Based on this, we define a total order $\prec_{pr}$ to prioritize the agents in the bag filling algorithm.

Definition 3.31 *Define a total order $\prec_{pr}$ on the agents of $\mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3$ with the following rules:*

(i). $a_{i_5} \prec_{pr} a_{i_1} \prec_{pr} a_{i_2} \prec_{pr} a_{i_3} \prec_{pr} a_{i_4}$ $\quad \forall a_{i_1} \in \mathcal{C}_1, a_{i_2} \in \mathcal{C}_2, a_{i_3} \in \mathcal{C}_3^s, a_{i_4} \in \mathcal{C}_3^b, a_{i_5} \in \mathcal{C}_3^f$

(ii). $a_i \prec_{pr} a_j \Leftrightarrow a_i \prec_o a_j$ $\quad \forall a_i, a_j \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3^s, a_i, a_j$ *in the same cluster*

(iii). $a_i \prec_{pr} a_j \Leftrightarrow i < j$ $\quad \forall a_i, a_j \in \mathcal{C}_3^b \vee a_i, a_j \in \mathcal{C}_3^f$

Recall that $\prec_o$ refers to the topological ordering of a semi-satisfied set of agents. Roughly speaking, for the semi-satisfied agents in the same cluster, $\prec_{pr}$ behaves in the same way as $\prec_o$. Furthermore, for the agents in different clusters, agents in $\mathcal{C}_3^f, \mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3^s, \mathcal{C}_3^b$ have a lower priority, respectively. Finally, the order of the agents in $\mathcal{C}_3^b$ and $\mathcal{C}_3^f$ is determined by their index, i.e., the agent with a lower index has a lower priority.

The second phase consists of several rounds and every round has two steps. Each of these two steps is described below. We continue running this algorithm until $\mathcal{F}$ is no longer feasible for any agent.

- **Step1**: In the first step, we run a process very similar to the bag filling algorithm described in Section 1. That is, we find a feasible subset $S \subseteq \mathcal{F}$, such that $|S|$ is minimal. Such a subset can easily be found, using a slight modification of the bag filling process (see Section 3.6.2).

- **Step2**: In the second step, we choose an agent to allocate set S to him. In contrast to the bag filling algorithm, we do not select an arbitrary agent. Instead, we select the agent in $\Phi(S)$ with the lowest priority regarding $\prec_{pr}$, i.e., smallest element in $\Phi(S)$ regarding $\prec_{pr}$. Let a_i be the selected agent. We consider three cases separately:

  (i). $a_i \in \mathcal{C}_3^f$: temporarily allocate S to a_i, i.e., set $f_i = S$.

  (ii). $a_i \in \mathcal{C}_3^b$: let a_j be the agent that $V_i(f_j) = 3/4 - \epsilon_j$. Take back f_j from a_j and allocate $f_j \cup S$ to a_i i.e. set $f_i = f_j$, $f_j = \emptyset$ and $g_i = S$. Remove a_i from $\mathcal{C}_3$ and add it to $\mathcal{S}$.

  (iii). $a_i \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3^s$: satisfy agent a_i by S, i.e., set $g_i = S$ and remove a_i from its corresponding cluster and add it to $\mathcal{S}$.

  By the construction of $\mathcal{C}_3^s, \mathcal{C}_3^b$, and $\mathcal{C}_3^f$, the above process may cause agents in $\mathcal{C}_3$ to move in between $\mathcal{C}_3^s, \mathcal{C}_3^b$ and $\mathcal{C}_3^f$. For example, if the first case happens, then a_i is moved from $\mathcal{C}_3^f$ to $\mathcal{C}_3^s$. In addition, all other agents in $\mathcal{C}_3^f$ for which S is feasible are moved to $\mathcal{C}_3^b$. For the second case, a_j is moved to one of $\mathcal{C}_3^f$ or $\mathcal{C}_3^b$, based on $V_j(f_k)$ for every $a_k \in \mathcal{C}_3^s$; that is, if there exists an agent $a_k \in \mathcal{C}_3^s$ such that $V_j(f_k) \geq 1/2$, a_j is moved to $\mathcal{C}_3^b$. Otherwise, a_j is moved to $\mathcal{C}_3^f$. For both the second and the third cases, some of the agents in $\mathcal{C}_3^b$ may move to $\mathcal{C}_3^f$.

The second phase terminates, when $\mathcal{F}$ is no longer feasible for any agent. More details about the second phase can be found in Algorithm 3. In Algorithm 3, we use $Update(\mathcal{C}_3)$ to refer the process of moving agents among $\mathcal{C}_3^s, \mathcal{C}_3^b$ and $\mathcal{C}_3^f$.

In each round of the second phase, either an agent is satisfied or an agent in $\mathcal{C}_3^f$ becomes semi-satisfied. In Lemma 3.32, we show that if an agent $a_i \in \mathcal{C}_3^f$ is selected in some round of the second phase, then $V_j(f_i)$ is upper bounded by $2\epsilon_j$ for every agent $a_j \in \mathcal{C}_3 \cup \mathcal{C}_2 \cup \mathcal{C}_1^s \cup \mathcal{C}_1^b$. As a consequence of Lemma 3.32, in Lemma 3.33 we show that sets $\mathcal{C}_1, \mathcal{C}_2$ and $\mathcal{C}_3$ remain cycle-envy-free during the second phase. For convenience, we use $\mathbb{R}_z$ to refer to the z'th round of the second phase.

Algorithm 3: The Second Phase

Data: $\mathcal{F}, \mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$

while $\mathcal{F}$ *is feasible* **do**
- $S =$ a minimal feasible subset of $\mathcal{F}$;
- $a_i =$ agent in $\Phi(S)$ with lowest order regarding $\prec_{pr}$;
- **if** $a_i \in C_3^f$ **then**
  - $f_i = S$;
  - $Update(\mathcal{C}_3)$;
- **if** $a_i \in \mathcal{C}_3^b$ **then**
  - Let a_j be the agent that $V_i(f_j) = 3/4 - \epsilon_i$;
  - $f_i = f_j$;
  - $g_i = S$;
  - $\mathcal{S} = \mathcal{S} \cup a_i$;
  - $f_j = \emptyset$;
  - $\mathcal{C}_3 = \mathcal{C}_3 \setminus a_i$;
  - $Update(\mathcal{C}_3)$;
- **if** $a_i \in \mathcal{C}_3^s$ **then**
  - $g_i = S$;
  - $\mathcal{S} = \mathcal{S} \cup a_i$;
  - $\mathcal{C}_3 = \mathcal{C}_3 \setminus a_i$;
  - $Update(\mathcal{C}_3)$;
- **if** $a_i \in \mathcal{C}_1 \cup \mathcal{C}_2$ **then**
  - $g_i = S$;
  - remove a_i from its corresponding cluster ;
  - $\mathcal{S} = \mathcal{S} \cup a_i$;

Lemma 3.32 *Let $\mathbb{R}_z$ be a round of the second phase that an agent $a_i \in \mathcal{C}_3^f$ is selected. Then, for every agent $a_j \in \mathcal{C}_3 \cup \mathcal{C}_2 \cup \mathcal{C}_1^s \cup \mathcal{C}_1^b$, we have $V_j(f_i) < 2\epsilon_j < 3/4$.*

Lemma 3.33 *During the second phase, the $\mathcal{C}_1, \mathcal{C}_2$ and $\mathcal{C}_3^s$ maintain the property of cycle-envy-freeness.*

Finally, for the rounds that an agents a_i is satisfied, Lemmas 3.34 and 3.35 give upper bounds on the value of g_i for remaining agents in different clusters.

Lemma 3.34 (value-lemma) *Let $a_i \in \mathcal{S}$ be an agent that is satisfied in the second phase. Then, for every other agent $a_j \in \mathcal{C}_1 \cup \mathcal{C}_2$ we have:*

(i). If $a_j \prec_{pr} a_i$, then $V_j(g_i) < \epsilon_j$.

(ii). If $a_i \prec_{pr} a_j$, then $V_j(g_i) < 2\epsilon_j$.

Lemma 3.35 (value-lemma) *Let a_i be an agent in $\mathcal{S}_1^s \cup \mathcal{S}_2^s$. Then, for every agent $a_j \in \mathcal{C}_3$, we have $V_j(g_i) < 1/2$.*

The results of Lemmas 3.34 and 3.35 are summarized in Table 5.

Table 5: Summary of value lemmas for g_i

	$\forall a_i \in \mathcal{C}_1$	$\forall a_i \in \mathcal{C}_2$	$\forall a_i \in \mathcal{C}_3$
$\forall a_j \in \mathcal{S}_1^s$	-	$V_i(g_j) < 2\epsilon_i(\star)$	$V_i(g_j) < 1/2(\dagger)$
$\forall a_j \in \mathcal{S}_2^s$	$V_i(g_j) < \epsilon_i(\star)$	-	$V_i(g_j) < 1/2(\dagger)$
$\forall a_j \in \mathcal{S}_3$	$V_i(g_j) < \epsilon_i(\star)$	$V_i(g_j) < \epsilon_i(\star)$	-

$\star$: Lemma 3.34 $\dagger$: Lemma 3.35

3.5 The Algorithm Finds a 3/4-MMS Allocation

In the rest of this section, we prove that the algorithm finds a 3/4-MMS allocation. For the sake of contradiction, suppose that the second phase is terminated, which means $\mathcal{F}$ is not feasible anymore, but not all agents are satisfied. Such an unsatisfied agent belongs to one of the Clusters $\mathcal{C}_1$ or $\mathcal{C}_2$, or $\mathcal{C}_3$. In Lemmas 3.36, 3.37, and 3.38, we separately rule out each of these possibilities. This implies that all the agents are satisfied and contradicts the assumption. For brevity the proofs are omitted and included in Appendix E. We begin with Cluster $\mathcal{C}_3$.

Lemma 3.36 *At the end of the algorithm we have $\mathcal{C}_3 = \emptyset$.*

To prove Lemma 3.36 we consider two cases separately. If $\mathcal{C}_3 \neq \emptyset$, either there exists an agent $a_i \in \mathcal{C}_3^s \cup \mathcal{C}_3^b$ or all the agents of $\mathcal{C}_3$ are in $\mathcal{C}_3^f$. If the former holds, we show $\mathcal{C}_3^s$ is non-empty and assume a_i is a winner of $\mathcal{C}_3^s$. We bound the total value of a_i for all the items dedicated to other agents and show the value of the remaining items in $\mathcal{F}$ is at least ϵ_i for a_i. This shows set $\mathcal{F}$ is feasible for a_i and contradicts the termination of the algorithm. In case all agents of $\mathcal{C}_3$ are in $\mathcal{C}_3^f$, let a_i be an arbitrary agent of $\mathcal{C}_3^f$. With a similar argument we show that the value of a_i for the remaining unassigned items is at least 3/4 and conclude that $\mathcal{F}$ is feasible for a_i which again contradicts the termination of the algorithm.

Next, we prove a similar statement for $\mathcal{C}_1$.

Lemma 3.37 *At the end of the algorithm we have $\mathcal{C}_1 = \emptyset$.*

Proof of Lemma 3.37 follows from a coloring argument. Let a_i be a winner of $\mathcal{C}_1$. We color all items in either blue or white. Roughly speaking, blue items are in a sense *heavy*, i.e., they may have a high valuation to a_i whereas white items are somewhat *lighter* and have a low valuation to a_i. Next, via a double counting argument, we show that a_i's value for the items of $\mathcal{F}$ is at least ϵ_i and thus $\mathcal{F}$ is feasible for a_i. This contradicts $\mathcal{C}_1 = \emptyset$ and shows at the end of the algorithm all agents of $\mathcal{C}_1$ are satisfied.

Finally, we show that all the agents in Cluster $\mathcal{C}_2$ are satisfied by the algorithm.

Lemma 3.38 *At the end of the algorithm we have $\mathcal{C}_2 = \emptyset$.*

The proof of Lemma 3.38 is a similar to both proofs of Lemmas 3.36 and 3.37. Let a_i be winner of Cluster $\mathcal{C}_2$. We consider two cases separately. (i) $\epsilon_i \geq 1/8$ and (ii) $\epsilon_i < 1/8$. In case $\epsilon_i \geq 1/8$, we use a similar argument to the proof of Lemma 3.36 and show $\mathcal{F}$ is feasible for a_i. If $\epsilon_i < 1/8$ we again use a coloring argument, but this time we color the items with 4 different colors. Again, via a double counting argument we show $\mathcal{F}$ is feasible for a_i and hence every agent of $\mathcal{C}_2$ is satisfied when the algorithm terminates.

Theorem 3.39 *All the agents are satisfied before the termination of the algorithm.*

Proof. By Lemmas 3.36, 3.37, and 3.38, at the end of the algorithm all agents are satisfied which means each has received a subset of items which is worth at least 3/4 to him. □

3.6 Algorithm

In this section, we present a polynomial time algorithm to find a $(3/4 - \epsilon)$-MMS allocation in the additive setting. More precisely, we show that our method for proving the existence of a 3/4-MMS allocation can be used to find such an allocation in polynomial time. Recall that our algorithm consists of two main phases: The clustering phase and the bag filling phase. In Sections 3.6.1 and 3.6.2 we separately explain how to implement each phase of the algorithm in polynomial time. Given this, there are still a few computational issues that need to be resolved. First, in the existential proof, we assume $\mathsf{MMS}_i = 1$ for every agent $a_i \in \mathcal{N}$. Second, we assume that the problem is 3/4-irreducible. Both of these assumptions are without loss of generality for the existential proof due to Observation 2.1 and the fact that one can scale the valuation functions to ensure $\mathsf{MMS}_i = 1$ for every agent a_i. However, the computational aspect of the problem will be affected by these assumptions. The first issue can be alleviated by incurring an additional $1 + \epsilon$ factor to the approximation guarantee. Epstein and Levin [22] show that for a given additive function f, MMS^n_f can be approximated within a factor $1 + \epsilon$ for constant ϵ in time $\mathsf{poly}(n)$. Thus, we can scale the valuation functions to ensure $\mathsf{MMS}_i = 1$ while losing a factor of at most $1 + \epsilon$. Therefore, finding a $(3/4 - \epsilon)$-MMS allocation can be done in polynomial time if the problem is 3/4-irreducible. Finally, in Section 3.6.3 we show how to reduce the 3/4-reducible instances and extend the algorithm to all instances of the problem. The algorithm along with the reduction yields Theorem 3.40

Theorem 3.40 *For any $\epsilon > 0$, there exists an algorithm that finds a $(3/4 - \epsilon)$-MMS allocation in polynomial time.*

3.6.1 The Clustering Phase

Recall that in the clustering phase we cluster the agents into three sets $\mathcal{C}_1$,$\mathcal{C}_2$, and $\mathcal{C}_3$. In order to build Cluster $\mathcal{C}_1$, we find an MCMWM of the 1/2-filtering of the value graph. This can be trivially done in polynomial time since finding an MCMWM is polynomially tractable [51]. However, the refinement phase of Cluster $\mathcal{C}_1$ requires finding $F_G(\mathcal{X}, M)$ for a giving graph G and a matching M. In what follows, we show this problem can also be solved in polynomial time.

Notice that finding an MCMWM of G can be done in polynomial time [51]. Therefore, in order to determine $F_H(M, \hat{\mathcal{X}})$, it only suffices to find the vertices of $\hat{\mathcal{X}}$ that are reachable from the unmatched vertices of $\hat{\mathcal{Y}}$ by an alternating path. Let $\hat{X}$ be the set of these vertices. We can find $\hat{X}$ using a depth-first-search from the unmatched vertices of $\hat{\mathcal{Y}}$. By definition, $F_H(M, \hat{\mathcal{X}}) = \hat{\mathcal{Y}} \setminus \hat{X}$. Therefore, $F_H(M, \hat{\mathcal{X}})$ can be found in polynomial time.

In addition to $F_G(\mathcal{X}, M)$, we also need to find a matching of the graph which satisfies the conditions of Lemma 3.15. We show in the following that this problem also can be solved in polynomial time. First, note that in Lemma C.1 we prove that G_1 has a matching that saturates all the vertices of W_1. Now, let p_{a_k} be the position of a_k in the topological ordering of $\mathcal{C}_1$, as described in the proof of Lemma 3.15. Furthermore, Let M_1 be a matching that minimizes the following expression.

$$\sum_{(x_j, y_i) \in M_1} p_i.$$

Recall that in the proof Lemma 3.15, we show that M_1 satisfies the condition described in Lemma 3.15. Here, we show that M_1 can be found in polynomial time. To this end, we model this with a network design problem.

Orient every edge $(x_j, y_i) \in G_1$ from y_i to x_j and set the cost of this edge to p_{a_i}. Also, add a source node s and add a directed edge from s to every vertex of $V_{\mathcal{C}_1}$ with cost 0. Furthermore, add a sink node t and add directed edges from the vertices of W_1 to t with cost 0. Finally, set the capacity of all edges to 1.

One can observe that in a minimum cost maximum flow from s to t in this network, the edges with non-zero flow between $V_{\mathcal{C}_1}$ and W_1 form a maximum matching M_1. In addition to this, since the cost of the flow is minimal, $\sum_{(x_j,y_i)\in M_1} cost(x_j, y_i)$ is minimized. Therefore, in this matching, $\sum_{(x_j,y_i)\in M_1} p_i$ is minimized. Thus, the matching with desired properties of Lemma 3.15 can be found in polynomial time.

The same algorithms can be used to compute Cluster $\mathcal{C}_2$. Finally, we put the rest of the agents in Cluster $\mathcal{C}_3$.

3.6.2 The bag filling Phase

In each round of the second phase, we iteratively find a minimal feasible subset of $\mathcal{F}$ and allocate its items to the agent with the lowest priority in $\Phi(S)$. Note that for a feasible set S, one can trivially find the agent with lowest priority in $\Phi(S)$ in polynomial time. Thus, it only remains to show that we can find a minimal feasible subset of $\mathcal{F}$ in polynomial time.

Consider the following algorithm, namely *reverse bag filling algorithm*: Start with a bag containing all the items of $\mathcal{F}$ and so long as there exists an item b_j in the bag such that after removing b_j, the set of items in the bag is still feasible, remove b_j from the bag. After this process, the remaining items in the bag form a minimally feasible subset of $\mathcal{F}$. Therefore, this phase can be run in polynomial time.

3.6.3 Reducibility

The most challenging part of our algorithm is dealing with the 3/4-irreducibility assumption. The catch is that, in order to run the algorithm, we don't necessarily need the 3/4-irreducibility assumption. Recall that we leverage the following three consequences of irreducibility to prove the existential theorem.

- The value of every item in $\mathcal{M}$ is less that 3/4 to every agent.
- Every pair of items in $\mathcal{X}'' \setminus \mathcal{X}''_{1/2}$ is in total worth less than 3/4 to any agent.
- The condition of Lemma C.1 holds.

Therefore, the algorithm works so long as the mentioned conditions hold. Note that, although it is not clear whether determining if an instance of the problem is 3/4-reducible is polynomially tractable, all of the above conditions can be validated in polynomial time. This is trivial for the first two conditions; we iterate over all items or pairs of items and check if the condition holds for these items. The last condition, however, is harder to validate.

The condition of Lemma C.1 holds if for all $S \subseteq W_1$, $|N(S)| > |S|$. Recall that in the proof of Lemma C.1 we showed that if this condition does not hold, then $F_{G_1}(M, \mathcal{X})$ is non-empty. Next, we showed that if $F_{G_1}(M, \mathcal{X})$ is non-empty, then we can reduce the problem via satisfying every agents of $F_{G_1}(M, \mathcal{X})$ by his matched item in M. Therefore, on the computational side, we only need to find whether $F_{G_1}(M, \mathcal{X})$ is empty which indeed can be determined in polynomial time.

Note that every time we reduce the problem, $|\mathcal{N}|$ is decreased by at least 1, which implies the number of times we reduce the problem is no more than n. Moreover, our reduction takes a polynomial time. Thus, the running time of the algorithm is polynomial.

4 Submodular Agents

Previous work on the fair allocation problem was limited to the additive agents [2, 46]. In real-world, however, valuation functions are usually more complex than additive ones. As an example, imagine an agent is interested in at most k items. More precisely, he is indifferent between receiving k items or more than k items. Such a valuation function is called k-demand and cannot be modeled by additive functions. k-demand functions are a subclass of submodular set functions which have been extensively studied the literature of different contexts, e.g., optimization, mechanism design, and game theory [13, 14, 32, 35, 37, 38, 40, 45, 50].

In this section, we study the fair allocation problem where the valuations of agents are submodular. We begin by presenting an impossibility result; We show in Section 4.1 that the best guarantee that we can achieve for submodular agents is upper bounded by 3/4. Next, we give a proof to the existence of a 1/3-MMS allocation in this setting. This is followed by an algorithm that finds such an allocation in polynomial time. This is surprising since even finding the MMS of a submodular function is NP-hard and cannot be implemented in polynomial time unless P=NP [22]. In our algorithm, we assume we have access to query oracle for the valuation of agents; That is, for any set S and any agent a_i, $V_i(S)$ can be computed via a given query oracle in time $O(1)$.

4.1 Upper Bound

We begin by providing an upper bound. In this section, we show for some instances of the problem with submodular agents, no allocation can be better than 3/4-MMS. Our counter-example is generic; We show this result for any number of agents.

Theorem 4.1 *For any $n \geq 2$, there exists an instance of the fair allocation problem with n submodular agents where no allocation is better than* 3/4-MMS.

Proof. We construct an instance of the problem that does not admit any $3/4 + \epsilon$-MMS allocation. To this end, let n be the number of agents and $\mathcal{M} = \{b_1, b_2, \ldots, b_m\}$ where $m = 2n$. Furthermore, let $f : 2^{\mathcal{M}} \to \mathbb{R}$ be as follows:

$$f(S) = \begin{cases} 0, & \text{if } |S| = \emptyset \\ 1, & \text{if } |S| = 1 \\ 2, & \text{if } |S| > 2 \\ 2, & \text{if } S = \{b_{2i}, b_{2i+1}\} \text{ for some } i \\ 3/2, & \text{if } |S| = 2 \text{ and } S \neq \{b_{2i}, b_{2i+1}\} \text{ for any } i. \end{cases}$$

Notice that $\mathsf{MMS}^n_f = 2$. Moreover, in what follows we show that f is submodular. To this end, suppose for the sake of contradiction that there exist sets S and S' such that $S \subseteq S'$ and for some element b_i we have:

$$f(S' \cup \{b_i\}) - f(S') > f(S \cup \{b_i\}) - f(S). \tag{3}$$

Since f is monotone and $S' \neq S$, $f(S' \cup \{b_i\}) - f(S') > 0$ holds and thus S' cannot have more than two items. Therefore, S' contains at most two items and thus S is either empty or contains a single element. If S is empty, then adding every element to S has the highest increase in the value

of S and thus Inequality (3) doesn't hold. Therefore, S contains a single element and S' contains exactly two elements. Thus, $f(S) = 1$ and $f(S') \geq 3/2$. Therefore, f(S $\cup\{b_i\}) - f(S) \geq 1/2$ and $f(S' \cup \{b_i\}) - f(S') \leq 1/2$ which contradicts Inequality (3).

Now, for agents $a_1, a_2, \ldots, a_{n-1}$ we set $V_i = f$ and for agent a_n we set $V_n = f(\mathsf{inc}(S))$ where b_i is in $\mathsf{inc}(S)$ if and only if either $i > 1$ and $b_{i-1} \in S$ or $i = 1$ and $b_m \in S$.

The crux of the argument is that for any allocation of the items to the agents, someone receives a value of at most 3/2. In case an agent receives fewer than two items, his valuation for his items would be at most 1. Similarly, if an agent receives more than two items, someone has to receive fewer than 2 items and the proof is complete. Therefore, the only case to investigate is where everybody receives exactly two items. We show in such cases, $\min V_i(A_i) = 3/2$ for all possible allocations. If all agents $a_1, a_2, \ldots, a_{n-1}$ receive two items whose value for them is exactly equal to 2, then by the construction of f, the value of the remaining items is also equal to 2 to them. Thus, a_n's valuation for the items he receives is equal to 3/2. □

Remark that one could replace function f with an XOS function

$$g(S) = \begin{cases} 0, & \text{if } |S| = \emptyset \\ 1, & \text{if } |S| = 1 \\ 2, & \text{if } |S| > 2 \\ 2, & \text{if } S = \{b_{2i}, b_{2i+1}\} \text{ for some } i \\ 1, & \text{if } |S| = 2 \text{ and } S \neq \{b_{2i}, b_{2i+1}\} \text{ for any } i. \end{cases}$$

and make the same argument to achieve a 1/2-MMS upper bound for XOS and subadditive agents.

Theorem 4.2 *For any $n > 1$, there exists an instance of the fair allocation problem with n XOS agents where no allocation is better than* 1/2-MMS*.*

4.2 Existential Proof

In this section we provide an existential proof to a 1/3-MMS allocation. Due to the algorithmic nature of the proof, we show in Section 4.3 that such an allocation can be computed in time $\mathsf{poly}(n, m)$. For simplicity, we scale the valuation functions to ensure $\mathsf{MMS}_i = 1$ for every agent a_i.

We begin by introducing the ceiling functions.

Definition 4.3 *Given a set function $f(.)$, we define $f^x(.)$ as follows:*

$$f^x(S) = \begin{cases} f(S), & \text{if } f(S) \leq x \\ x, & \text{if } f(S) > x. \end{cases}$$

A nice property of the ceiling functions is that they preserve submodularity, fractionally subadditivity, and sub-additivity as we show in Appendix F.

Lemma 4.4 *For any real number $x \geq 0$, we have:*

(i). Given a submodular set function $f(.)$, $f^x(.)$ is submodular.

(ii). Given an XOS set function $f(.)$, $f^x(.)$ is XOS.

(iii). Given an subadditive set function $f(.)$, $f^x(.)$ is also subadditive.

The idea behind the existence of a 1/3-MMS allocation is simple: Suppose the problem is 1/3-irreducible and let $\mathcal{A} = \langle A_1, A_2, \ldots, A_n \rangle$ be an allocation of items to the agents that maximizes the following expression:

$$\sum_{a_i \in \mathcal{N}} V_i^{2/3}(A_i) \tag{4}$$

We refer to Expression (4) by $\mathsf{ex}^{(2/3)}(\mathcal{A})$. We prove $V_i(A_i) \geq 1/3$ for every agent $a_i \in \mathcal{N}$. By the reducibility principal, it only suffices to show every 1/3-irreducible instance of the problem admits a 1/3-MMS allocation. The main ingredients of the proof are Lemmas 3.1, 4.5 and 4.6. For brevity we skip the proofs and include them in Appendix F.

Lemma 4.5 *Let $S_1, S_2, \ldots, S_k$ be k disjoint sets and $f_1, f_2, \ldots, f_k$ be k submodular functions. We remove an element e from $\bigcup S_i$ uniformly at random to obtain sets $S_1^* = S_1 \setminus \{e\}, S_2^* = S_2 \setminus \{e\}, \ldots, S_k^* = S_k \setminus \{e\}$. In this case we have*

$$\mathbb{E}[\sum f_i(S_i^*)] \geq \sum f_i(S_i) \frac{|\bigcup S_i| - 1}{|\bigcup S_i|}.$$

The high-level intuition behind the proof of Lemma 4.5 is as follows: For submodular functions, the smaller the size of a set is, the higher the marginal values for adding items to that set will be. Based on that, we show the summation of marginal decreases for removing each element is bounded by the total value of the set and that completes the proof. A complete proof is included in Appendix F.

Lemma 4.6 *Let f be a submodular function and $S_1, S_2, \ldots, S_k$ be k disjoint sets such that $f(S_i) \geq 1$ for every set S_i. Moreover, let $S \subseteq \bigcup S_i$ be a set such that $f(S) < 1/3$. If we pick an element $\{e\}$ of $\bigcup S_i \setminus S$ uniformly at random, we have:*

$$\mathbb{E}[f(S \cup \{e\}) - f(S)] \geq \frac{2k/3}{|\bigcup S_i \setminus S|}.$$

The proof of Lemma 4.6 is very similar to that of Lemma 4.5. The main point is that in submodular functions, the marginal increase decreases as the sizes of sets grow.

Next, we show the fair allocation problem with submodular agents admits a 1/3-MMS allocation[7].

Theorem 4.7 *The fair allocation problem with submodular agents admits a 1/3-MMS allocation.*

Proof. By Lemma 2.1, the problem boils down to the case of 1/3-irreducible instances. Let the problem be 1/3-irreducible and $\mathcal{A}$ be an allocation that maximizes $\mathsf{ex}^{(2/3)}$. Suppose for the sake of contradiction that $V_i(A_i) < 1/3$ for some agent a_i. In this case we select an item b_r from $\mathcal{M} \setminus A_i$ uniformly at random to create a new allocation $\mathcal{A}^r$ as follows:

$$A_j^r = \begin{cases} A_j \setminus \{b_r\}, & \text{if } i \neq j \\ A_j \cup \{b_r\} & \text{if } i = j. \end{cases}$$

[7]Almost one year after the first draft of our work, the existense of a 1/10-MMS allocation in the submodular case along with an algorithm to find a 1/31 approximation algorithm for the submodular case is also proved in [7]. They also study the problem in the additive setting and present another 2/3-MMS algorithm. This work is completely parallel to and independent of our paper. Moreover, their analysis is fundamentally different from our analysis and also their bounds are looser.

In the rest we show $\mathbb{E}[\mathsf{ex}^{(2/3)}(\mathcal{A}^r)] > \mathsf{ex}^{(2/3)}(\mathcal{A})$ which contradicts the maximality of $\mathcal{A}$. Note that by Lemma 4.5 the following inequality holds:

$$\mathbb{E}[\sum_{j\neq i} V_j^{2/3}(A_j^r)] \geq \sum_{j\neq i} V_j^{2/3}(A_j)\frac{|\mathcal{M}\setminus A_i|-1}{|\mathcal{M}\setminus A_i|}. \tag{5}$$

Moreover, by Lemma 4.6 we have

$$\mathbb{E}[V_i(A_i^r) - V_i(A_i)] \geq \frac{2n/3}{|\mathcal{M}\setminus A_i|}. \tag{6}$$

Inequality (5) along with Inequality (6) shows

$$\begin{aligned}
\mathbb{E}[\mathsf{ex}^{(2/3)}(\mathcal{A}^r)] &= \mathbb{E}[\sum_{j\neq i} V_j^{2/3}(A_j^r)] + \mathbb{E}[V_i(A_i^r)] \\
&\geq \sum_{j\neq i} V_j^{2/3}(A_j)\frac{|\mathcal{M}\setminus A_i|-1}{|\mathcal{M}\setminus A_i|} + \mathbb{E}[V_i(A_i^r)] \\
&\geq \sum_{j\neq i} V_j^{2/3}(A_j)\frac{|\mathcal{M}\setminus A_i|-1}{|\mathcal{M}\setminus A_i|} + \frac{2n/3}{|\mathcal{M}\setminus A_i|} + V_i(A_i) \\
&\geq \sum_{j\neq i} V_j^{2/3}(A_j)\frac{|\mathcal{M}\setminus A_i|-1}{|\mathcal{M}\setminus A_i|} + \frac{2n/3}{|\mathcal{M}\setminus A_i|} + V_i^{(2/3)}(A_i) \\
&\geq \sum_{j\neq i} V_j^{2/3}(A_j)\frac{|\mathcal{M}\setminus A_i|-1}{|\mathcal{M}\setminus A_i|} + \frac{2n/3}{|\mathcal{M}\setminus A_i|} + V_i^{(2/3)}(A_i)\frac{|\mathcal{M}\setminus A_i|-1}{|\mathcal{M}\setminus A_i|} \\
&= \mathsf{ex}^{(2/3)}(\mathcal{A})\frac{|\mathcal{M}\setminus A_i|-1}{|\mathcal{M}\setminus A_i|} + \frac{2n/3}{|\mathcal{M}\setminus A_i|}.
\end{aligned} \tag{7}$$

Recall that by Lemma 3.1, the value of agent a_i for any item alone is bounded by $1/3$ and thus $\mathbb{E}[V_i(A_i^r) - V_i(A_i)] = \mathbb{E}[V_i^{2/3}(A_i^r) - V_i^{2/3}(A_i)]$. Notice that by the definition, $V_j^{(2/3)}$ is always bounded by $2/3$ and also $V_i(A_i) < 1/3$, therefore, $\mathsf{ex}^{(2/3)}(\mathcal{A}) \leq 2n/3 - 1/3$ and thus

$$\begin{aligned}
\mathbb{E}[\mathsf{ex}^{(2/3)}(\mathcal{A}^r)] &\geq \mathsf{ex}^{(2/3)}(\mathcal{A})\frac{|\mathcal{M}\setminus A_i|-1}{|\mathcal{M}\setminus A_i|} + \frac{2n/3}{|\mathcal{M}\setminus A_i|} \\
&\geq \mathsf{ex}^{(2/3)}(\mathcal{A}) + \frac{1/3}{|\mathcal{M}\setminus A_i|} \\
&\geq \mathsf{ex}^{(2/3)}(\mathcal{A}) + 1/3m.
\end{aligned} \tag{8}$$

□

4.3 Algorithm

In this section we give an algorithm to find a 1/3-MMS allocation for submodular agents. We show our algorithm runs in time $\mathsf{poly}(n, m)$.

For simplicity, we assume for every agent a_i, MMS_i is given as input to the algorithm. However, computing MMS_i alone is an NP-hard problem. That said, we show in Section 5.2.2 that such a computational barrier can be lifted by a combinatorial trick. We refer the reader to Section 5.2.2 for a more detailed discussion. The procedure is illustrated in Algorithm 4: Based on Theorem

Algorithm 4: Finding a 1/3-MMS allocation for submodular agents

Data: $\mathcal{N}, \mathcal{M}, \langle V_1, V_2, \ldots, V_n \rangle, \langle \mathsf{MMS}_1, \mathsf{MMS}_2, \ldots, \mathsf{MMS}_n \rangle$

For every a_j, scale V_j to ensure $\mathsf{MMS}_j = 1$;
while *there exist an agent* a_i *and an item* b_j *such that* $V_i(\{b_j\}) \geq 1/3$ **do**
    Allocate $\{b_j\}$ to a_i;
    $\mathcal{M} = \mathcal{M} \setminus b_j$;
    $\mathcal{N} = \mathcal{N} \setminus a_i$;
$\mathcal{A} =$ an arbitrary allocation of the items to the agents;
while $\min V_j^{2/3}(A_j) < 1/3$ **do**
    $i =$ the agent who receives the lowest value in allocation $\mathcal{A}$;
    Find an item b_e such that:
    $\mathsf{ex}(\langle A_1 \setminus \{b_e\}, A_2 \setminus \{b_e\}, \ldots, A_{i-1} \setminus \{b_e\}, A_i \cup \{b_e\}, A_{i+1} \setminus \{b_e\}, \ldots, A_n \setminus \{b_e\} \rangle) \geq \mathsf{ex}(\mathcal{A}) + 1/3m$;
    $\mathcal{A} = \langle A_1 \setminus \{b_e\}, A_2 \setminus \{b_e\}, \ldots, A_{i-1} \setminus \{b_e\}, A_i \cup \{b_e\}, A_{i+1} \setminus \{b_e\}, \ldots, A_n \setminus \{b_e\} \rangle$;
For every $a_i \in \mathcal{N}$ allocate A_i to a_i;

4.7, one can show that in every iteration of the algorithm value of $\mathsf{ex}^{2/3}(\mathcal{A})$ is increased by at least $1/3m$. Moreover, such an element b_e can be easily found by iterating over all items in time $O(m)$. Furthermore, the number of iterations of the algorithm is bounded by $2nm$, since $\mathsf{ex}^{2/3}(\mathcal{A})$ is bounded by $2n/3$. Therefore, Algorithm 4 finds a 1/3-MMS allocation in time $\mathsf{poly}(n, m)$.

Theorem 4.8 *Given access to query oracles, one can find a* 1/3-MMS *allocation for submodular agents in polynomial time.*

As a corollary of Theorem 4.8, one can show that the problem of finding the maxmin value of a submodular function admits a 3 approximation algorithm.

Corollary 4.9 *For a given submodular function f, we can in polynomial time split the elements of ground set into n dijsoint sets $S_1, S_2, \ldots, S_n$ such that*

$$f(S_i) \geq \mathsf{MMS}_f^n/3$$

for every $1 \leq i \leq n$.

5 XOS Agents

Class of fractionally subadditive (XOS) set functions is a super class of submodular functions. These functions too, have been subject of many studies in recent years [18, 9, 25, 10, 49, 28, 31, 29, 44]. Similar to sub-modular functions, in this section we show a 1/5-MMS allocation is possible when all agents have XOS valuations. Furthermore, we complement our proof by providing a polynomial algorithm to find a 1/8-MMS allocation in Section 5.2.

5.1 Existential Proof

In this section we show every instance of the fair allocation problem with XOS agents admits a 1/5-MMS allocation. Without loss of generality, we assume $\mathsf{MMS}_i = 1$ for every agent a_i. Recall the definition of ceiling functions.

Definition 5.1 *Given a set function $f(.)$, we define $f^x(.)$ as follows:*

$$f^x(S) = \begin{cases} f(S), & \text{if } f(S) \leq x \\ x, & \text{if } f(S) > x. \end{cases}$$

As stated in Lemma 4.4, for every XOS function and every real number $x \geq 0$, f^x is also XOS. The proof of this section is similar to the result of Section 4. However, the details are different since XOS functions do not adhere to the nice structure of submodular functions. For every allocation $\mathcal{B}$, we define $\mathsf{ex}^{2/5}(\mathcal{B})$ as follows:

$$\mathsf{ex}^{2/5}(\mathcal{B}) = \sum_{a_i \in \mathcal{N}} V_i^{2/5}(B_i).$$

Now Let $\mathcal{A} = \langle A_1, A_2, \ldots, A_n \rangle$ be an allocation of items to the agents that maximizes $\mathsf{ex}^{2/5}$. Provided that the problem is 1/5-irreducible, we show $\mathcal{A}$ is a 1/5-MMS allocation. Before we proceed to the main proof, we state Lemmas 5.2, and 5.3 as auxiliary observations.

Lemma 5.2 *Let $f(.)$ be an XOS set function and $f(S) = \beta$ for a set $S \subseteq \mathsf{ground}(f)$. If we divide S into k (possibly empty) sets $S_1, S_2, \ldots, S_k$ then*

$$\sum_{i=1}^{k} \Big(f(S) - f(S \setminus S_i) \Big) \leq f(S).$$

The complete proof of Lemma 5.2 is included in Appendix G. Roughly speaking, the proof follows from the fact that for at least one of the additive set functions in the representation of f, we have $g_j(S) = \beta$. The rest of the proof is trivial by the additive properties of g_j.

By Lemma 3.1, we know that in every 1/5-irreducible instance of the problem, the value of every item for a person is bounded by 1/5. For XOS functions, we again, leverage the reducibility principal to show another important property of the 1/5-irreducible instances of the problem.

Lemma 5.3 *In a 1/5-irreducible instance of the problem, for a given agent a_i we can divide the items into $2n$ sets $S_1, S_2, \ldots, S_{2n}$ such that*

$$V_i(S_i) \geq 2/5$$

for every $1 \leq i \leq 2n$.

We first apply Lemma 3.1 and show in such instances of the problem the valuation of every agent for every item is bounded by 1/5. We remark that for every agent a_i, one can split the items into n partitions such that each partition is worth at least 1 to a_i. Combining the two observations, we conclude that such a decomposition is possible for every agent a_i. The full proof of this lemma is included in Appendix G. Next we prove the main theorem of this section.

Theorem 5.4 *The fair allocation problem with XOS agents admits a 1/5-MMS allocation.*

Proof. Similar to what we did in Section 1, we only prove this for 1/5-irreducible instances of the problem. By Observation 2.1, we can extend this result to all instances of the problem.

Consider an allocation $\mathcal{A} = \langle A_1, A_2, \ldots, A_n \rangle$ of items to the agents that maximizes $\mathsf{ex}^{2/5}$. We show that such an allocation is 1/5-MMS. Suppose for the sake of contradiction that there exists

an agent a_i who receives a set of items which are together of worth less than $1/5$ to him. More precisely,

$$V_i^{2/5}(A_i) = V_i(A_i) < 1/5.$$

Since the problem is $1/5$-irreducible, by Lemma 5.3, we can divide the items into $2n$ sets $S_1, S_2, \ldots, S_{2n}$ such that $V_i(S_j) \geq 2/5$ for every $1 \leq j \leq 2n$. Note that in this case, $V_i^{2/5}(S_j) = 2/5$ follows from the definition. Moreover by monotonicity, $V_i^{2/5}(S_j \cup A_i) = 2/5$ holds for every j.

Now consider $2n$ allocations $\mathcal{A}^1, \mathcal{A}^2, \ldots, \mathcal{A}^{2n}$ such that

$$\mathcal{A}^j = \langle A_1^j, A_2^j \ldots, A_n^j \rangle$$

for every $1 \leq j \leq 2n$ where

$$A_k^j = \begin{cases} A_k \cup S_j, & \text{if } k = i \\ A_k \setminus S_j, & \text{if } k \neq i. \end{cases}$$

We show at least one of these allocations has a higher for $\mathsf{ex}^{2/5}$ than $\mathcal{A}$. Since $V_i^{2/5}$ is XOS, by Lemma 5.2 we have

$$\sum_{j=1}^{2n} \left(V_k^{2/5}(A_k) - V_k^{2/5}(A_k \setminus S_j) \right) \leq V_k^{2/5}(A_j)$$

for every $a_k \neq a_i$ and thus

$$\begin{aligned} \sum_{j=1}^{2n} V_k^{2/5}(A_k^j) &= \sum_{j=1}^{2n} V_k^{2/5}(A_j \setminus S_j) \\ &\geq 2n V_k^{2/5}(A_k) - V_k^{2/5}(A_k) \\ &= (2n-1) V_k^{2/5}(A_k) \end{aligned} \tag{9}$$

Moreover, since $V_i^{2/5}(A_i) < 1/5$, we have

$$\begin{aligned} \sum_{a_j \neq a_i} V_j^{2/5}(A_j) &> \sum_{a_j \in \mathcal{N}} V_j^{2/5}(A_j) - 1/5 \\ &= \mathsf{ex}^{2/5}(\mathcal{A}) - 1/5. \end{aligned} \tag{10}$$

Furthermore, since $V_i^{2/5}(S_j \cup A_i) = 2/5$ for every $1 \leq j \leq 2n$, we have

$$\begin{aligned} \sum_{a_k \neq a_i} V_k^{2/5}(A_k^j) &= \sum_{a_k \in \mathcal{N}} V_k^{2/5}(A_k^j) - 2/5 \\ &= \mathsf{ex}^{2/5}(\mathcal{A}^j) - 2/5 \end{aligned} \tag{11}$$

Finally, by combining Inequalities (9), (10), and (11) we have

$$\begin{aligned}
\sum_{j=1}^{2n} \mathsf{ex}^{2/5}(\mathcal{A}^j) &= \sum_{j=1}^{2n}(2/5 + \sum_{a_k \neq a_i} V_k^{2/5}(A_k^j)) \\
&= 4n/5 + \sum_{j=1}^{2n} \sum_{a_k \neq a_i} V_k^{2/5}(A_k^j) \\
&\geq 4n/5 + \sum_{a_k \neq a_i} (2n-1) V_k^{2/5}(A_k) \\
&\geq 4n/5 + (2n-1)(\mathsf{ex}^{2/5}(\mathcal{A}) - 1/5) \\
&\geq 2n \cdot \mathsf{ex}^{2/5}(\mathcal{A}) + (4n - 2n + 1)/5 - \mathsf{ex}^{2/5}(\mathcal{A}) \\
&\geq 2n \cdot \mathsf{ex}^{2/5}(\mathcal{A}) + (2n+1)/5 - \mathsf{ex}^{2/5}(\mathcal{A})
\end{aligned}$$

Now notice that since $V_k^{2/5}(A_k) \leq 2/5$, we have

$$\begin{aligned}
\mathsf{ex}^{2/5}(\mathcal{A}) &= \sum_{k=1}^{n} V_k^{2/5}(A_k) \\
&\leq \sum_{k=1}^{n} 2/5 \\
&\leq 2n/5.
\end{aligned}$$

and thus

$$\begin{aligned}
\sum_{j=1}^{2n} \mathsf{ex}^{2/5}(\mathcal{A}^j) &\geq 2n \cdot \mathsf{ex}^{2/5}(\mathcal{A}) + (2n+1)/5 - \mathsf{ex}^{2/5}(\mathcal{A}) \\
&\geq 2n \cdot \mathsf{ex}^{2/5}(\mathcal{A}) + (2n+1)/5 - 2n/5 \\
&\geq 2n \cdot \mathsf{ex}^{2/5}(\mathcal{A}) + 1/5.
\end{aligned}$$

Therefore, $\mathsf{ex}^{2/5}(\mathcal{A}^j) > \mathsf{ex}^{2/5}(\mathcal{A}) + 1/10n$ holds for at least one $\mathcal{A}^j$ which contradicts the maximality of $\mathcal{A}$. □

5.2 Algorithm

In this section we provide a polynomial time algorithm for finding a 1/8-MMS allocation for the fair allocation problem with XOS agents. The algorithm is based on a similar idea that we argued for the proof of Theorem 5.4. Remark that our algorithm only requires access to demand and XOS oracles. It does *not* have any additional information about the maxmin values. This makes the problem computationally harder since computing the maxmin values is NP-hard [22]. We begin by giving a high-level intuition of the algorithm and show the computational obstacles can be overcome by combinatorial tricks. Consider the pseudo-code described in Algorithm 5.

As we show in Section 5.2.1, Command 9 of the algorithm is always doable. More precisely, there always exists a set S that holds in the condition of Command 9. Notice that in every step of the algorithm, $\mathsf{ex}^{1/4}(\mathcal{A})$ is increased by at least $1/12n$ and this value is bounded by $1/4 \cdot n = n/4$. Therefore the algorithm terminates after at most $3n^2$ steps and the allocation is guaranteed to be 1/8-MMS.

Algorithm 5: Algorithm for finding a 1/8-MMS allocation

Data: $\mathcal{N}, \mathcal{M}, \langle V_1, V_2, \ldots, V_n \rangle$

1 For every a_j, scale V_j to ensure $\mathsf{MMS}_j = 1$;
2 **while** *there exist an agent* a_i *and an item* b_j *such that* $V_i(\{b_j\}) \geq 1/8$ **do**
3 Allocate $\{b_j\}$ to a_i;
4 $\mathcal{M} = \mathcal{M} \setminus b_j$;
5 $\mathcal{N} = \mathcal{N} \setminus a_i$;
6 $\mathcal{A}$ = an arbitrary allocation of the items to the agents;
7 **while** $\min V_j^{1/4}(A_j) < 1/8$ **do**
8 i = the agent who receives the lowest value in allocation $\mathcal{A}$;
9 Find a set S such that:
$\mathsf{ex}^{1/4}(\langle A_1 \setminus S, A_2 \setminus S, \ldots, A_{i-1} \setminus S, A_i \cup S, A_{i+1} \setminus S, \ldots, A_n \setminus S \rangle) \geq \mathsf{ex}^{1/4}(\mathcal{A}) + 1/12n$;
10 $\mathcal{A} = \langle A_1 \setminus S, A_2 \setminus S, \ldots, A_{i-1} \setminus S, A_i \cup S, A_{i+1} \setminus S, \ldots, A_n \setminus S \rangle$;
11 For every $a_i \in \mathcal{N}$ allocate A_i to a_i;

That said, there are two major computational obstacles in the way of running Algorithm 5. Firstly, finding a set S that holds in the condition of Command 9 can not be trivially done in polynomial time. Second, scaling the valuation functions to ensure $\mathsf{MMS}_i = 1$ for all agents is NP-hard and cannot be done in polynomial time unless P=NP. To overcome the former, in Section 5.2.1 we provide an algorithm for finding such a set S in polynomial time. Next, in Section 5.2.2, we present a combinatorial trick to run the algorithm in polynomial time without having to deal with NP-hardness of scaling the valuation functions.

5.2.1 Executing Command 9 in Polynomial Time

In this section we present an algorithm to execute Command 9 of Algorithm 5. We show that such a procedure can be implemented via demand oracles.

Let for every $b_j \notin A_i$, c_j be the amount of contribution that b_j makes to $\mathsf{ex}^{1/4}(\mathcal{A})$. We set $p_e = 3(n/(n-1))c_e$ and ask the demand oracle of V_i to find a set S that maximizes $V_i(S) - \sum_{b_j \in S} p_j$. Via a trivial calculation, one can show that $V_i(S) - \sum_{b_j \in S} p_j \geq 1/4$ holds for at least one set of items. The reason this is correct is that one can divide the items into n partitions where each is worth at least 1 to a_i. Moreover, the summation of prices for the items is bounded by $3n/(n-1) \cdot (\sum_{j \neq i} V_j^{1/4}(A_j)) \leq 3n/4$. Therefore, for at least one of those partitions $V_i(S) - \sum_{b_j \in S} p_j$ is at least $1/4$. Thus, the set that the oracle reports is worth at least $1/4$ to a_i.

Now, let S^* be the set that the oracle reports and for every $b_j \in S^*$, c_j^* be the contribution of b_j to $V_i(S^*)$. We sort the items of S^* based on $c_j^* - p_j$ in non-increasing order. Next, we start with an empty bag and add the items in their order to the bag until the total value of the items in the bag to a_i reaches $1/4$. Since the value of every item alone is bounded by $1/8$, the total value of the items in the bag to a_i is bounded by $3/8$. Thus the contribution of those items to $\mathsf{ex}^{1/4}(\mathcal{A})$ is at most $(3/8)/(3n/(n-1)) \leq 1/8 - 1/(10n)$. Therefore, removing items of the bag from other allocations and adding them to A_i, increases $\mathsf{ex}^{1/4}(\mathcal{A})$ by at least $1/10n$.

Remark that one can use the same argument to prove this even if $\mathsf{MMS}_i \geq 1/(1 + 1/10n)$.

5.2.2 Running Algorithm 5 in Polynomial Time

As aforementioned, scaling valuation functions to ensure $\mathsf{MMS}_i = 1$ for every agent a_i is an NP-hard problem since determining the maxmin values is hard even for additive agents [46]. Therefore, unlike Section 5.2.1, in this section we massage the algorithm to make it executable in polynomial time.

Suppose an oracle gives us the maxnmin values of the agents. Provided that we can run Command 9 of Algorithm 5 in polynomial time, we can find a 1/8-MMS allocation in polynomial time. Therefore, in case the oracle reports the actual maxmin values, the solution is trivial. However, what if the oracle has an error in its calculations? There are two possibilities: (i) Algorithm 5 terminates and finds an allocation which is 1/8-MMS with respect to the reported maxmin values. (ii) The algorithm fails to execute Command 9, since no such set S holds in the condition of Command 9. The intellectual merit of this section boils down to investigation of the case when algorithm fails to execute Command 9. We show, this only happens due to an overly high misrepresentation of the maxmin value for agent a_i. Note that a_i is the agent who receives the lowest value in the last cycle of the execution.

Observation 5.1 *Given $\langle d_1, d_2, \ldots, d_n \rangle$ as an estimate for the maxmin values, if Algorithm 5 fails to execute Command 9 for an agent a_i, then we have*

$$d_i \geq (1 + 1/10n)\mathsf{MMS}_i.$$

Proof of Observation 5.1 follows from the argument of Section 5.2.1. More precisely, as mentioned in Section 5.2.1, such a set S exists, if $\mathsf{MMS}_i \geq 1/(1 + 1/10n)$. Thus, given that the procedure explained in Section 5.2.1 fails to find such a set, one can conclude the the reported value for MMS_i is at least $(1/(1 + 1/10n))$ times its actual value. Based on Observation 5.1, we propose Algorithm 6 for implementing a maxmin oracle.

Algorithm 6: Implementing a maxmin oracle

```
Data: 𝒩, ℳ, ⟨V_1, V_2, ..., V_n⟩
for every a_i ∈ 𝒩 do
    d_i ← V_i(ℳ);
while true do
    Run Algorithm 5 assuming maxmin values are d_1, d_2, ..., d_n;
    if the Algorithm fails to run Command 9 for an agent a_i then
        d_i ← d_i/(1 + 1/10n);
    else
        Report the allocation and terminate the algorithm;
```

Note that in the beginning of the algorithm, we set $d_i = V_i(\mathcal{M})$ which is indeed greater than or equal to MMS_i. By Lemma 5.1, every time we decrease the value of d_i for an agent a_i, we preserve the condition $d_i \geq \mathsf{MMS}_i$ for that agent. Therefore, in every step of the algorithm, we have $d_i \geq \mathsf{MMS}_i$ and thus the reported allocation which is 1/8-MMS with respect to d_i's is also 1/8-MMS with respect to true maxmin values. Thus, the algorithm provides a correct 1/8-MMS allocation in the end. All that remains is to show the running time of the algorithm is polynomial.

Notice that every time we decrease d_i for an agent a_i, we multiply this value by $1/(1 + 1/10n)$, hence the number of such iterations is polynomial in n, unless the valuations are super-exponential

in n. Since we always assume the input numbers are represented by $\mathsf{poly}(n)$ bits, the number of iterations is bounded by $\mathsf{poly}(n)$ and hence the algorithm terminates after a polynomial number of steps.

Theorem 5.5 *Given access to demand and XOS oracles, there exists a polynomial time algorithm that finds a* $1/8$-MMS *allocation for XOS agents.*

An elegant consequence of Theorem 5.5 is a 8-approximation algorithm for determining the maxmin value of an XOS function with r partitions.

Corollary 5.6 *Given an XOS function f, an integer number r, and access to demand and XOS oracles of f, there exists a 8-approximation polytime algorithm for determining MMS_f^r.*

Proof. We construct an instance of the fair allocation problem with r agents, all of whom have a valuation function equal to f. We find a $1/8$-MMS allocation of the items to the agents in polynomial time and report the minimum value that an agent receives as output.

The $1/8$ guarantee follows from the fact that every agent receives a subset of values that are worth $1/8$-MMS_i to him, and since MMS_i is exactly equal to MMS_f^r, every partition has a value of at least $\mathsf{MMS}_f^r/8$. □

Remark 5.7 *A similar procedure can also be used to overcome the challenge of computing the maxmin values for the algorithm described in Section 4.3.*

6 Subadditive Agents

In this section we present a reduction from subadditive agents to XOS agents. More precisely, we show for every subadditive set function $f(.)$, there exists an XOS function $g(.)$, where g is dominated by f but the maxmin value of g is within a logarithmic factor of the maxmin value of f. We begin by an observation. Suppose we are given a subadditive function f on set $\mathsf{ground}(f)$, and we wish to approximate f with an additive function g which is dominated by f. In other words, we wish to find an additive function g such that

$$\forall S \subseteq \mathsf{ground}(f) \qquad g(S) \leq f(S)$$

and $g(\mathsf{ground}(f))$ is maximized. One way to formulate g is via a linear program. Suppose $\mathsf{ground}(f) = \{b_1, b_2, \ldots, b_m\}$ and let $g_1, g_2, \ldots, g_m$ be m variables that describe g in the following way:

$$\forall S \subseteq \mathsf{ground}(f) \qquad g(S) = \sum_{b_i \in S} g_i.$$

Based on this formulation, we can find the optimal additive function g by LP 12.

$$\begin{aligned} \text{maximize:} \quad & \sum_{b_i \in \mathsf{ground}(f)} g_i & & (12) \\ \text{subject to:} \quad & \sum_{b_i \in S} g_i \leq f(S) & \forall S \subseteq \mathsf{ground}(f) & \\ & g_i \geq 0 & \forall b_i \in \mathsf{ground}(f) & \end{aligned}$$

We show the objective function of LP 12 is lower bounded by $f(\mathsf{ground}(f))/\log m$. The basic idea is to first write the dual program and then based on a probabilistic method, lower bound the optimal value of the dual program by $f(\mathsf{ground}(f))/\log m$.

Lemma 6.1 *The optimal solution of LP 12 is at least* $f(\mathsf{ground}(f))/\log m$.

Proof. To prove the lemma, we write the dual of LP 12 as follows:

$$\begin{aligned} \text{minimize:} \quad & \sum_{S \subseteq \mathsf{ground}(f)} \alpha_S f(S) && && (13)\\ \text{subject to:} \quad & \sum_{S \ni b_i} \alpha_S \geq 1 && \forall b_i \in \mathsf{ground}(f) \\ & \alpha_S \geq 0 && \forall S \subseteq \mathsf{ground}(f) \end{aligned}$$

By the strong duality theorem, the optimal solutions of LP 12 and LP 13 are equal [5]. Next, based on the optimal solution of LP 13, we define a randomized procedure to draw a set of elements: We start with an empty set S^* and for every set $S \subseteq \mathsf{ground}(f)$ we add *all* elements of S to S^* with probability α_S. Since f is subadditive, the marginal increase of $f(S^*)$ by adding elements of a set S to S^* is bounded by $f(S)$ and thus the expected value of $f(S^*)$ is bounded by the objective of LP 13. In other words:

$$\mathbb{E}[f(S^*)] \leq \sum_{S \subseteq \mathsf{ground}(f)} \alpha_S f(S) \tag{14}$$

Remark that we repeat this procedure for all subsets of $\mathsf{ground}(S)$ independently and thus for every $b_i \in \mathsf{ground}(f)$, $\sum_{S \ni b_i} \alpha_S \geq 1$ holds we have

$$\mathsf{PR}[b_i \in S^*] \geq 1 - 1/e \simeq 0.632121 > 1/2 \tag{15}$$

for every element $b_i \in \mathsf{ground}(s)$. Now, with the same procedure, we draw $\lceil \log m \rceil + 2$ sets $S_1^*, S_2^*, \ldots, S_{\lceil \log m \rceil + 2}^*$ *independently*. We define $\hat{S} = \bigcup S_i^*$. By Inequality (15) and the union bound we show

$$\begin{aligned} \mathsf{PR}[\hat{S} = \mathsf{ground}(f)] &\geq 1 - \sum_{b_i \in \mathsf{ground}(i)} \mathsf{PR}[b_i \notin \hat{S}] \\ &= 1 - \sum_{b_i \in \mathsf{ground}(i)} \mathsf{PR}[b_i \notin S_1^* \text{ and } b_i \notin S_1^* \text{ and } \ldots \text{ and } b_i \notin S_{\lceil \log m \rceil + 2}^*] \\ &= 1 - \sum_{b_i \in \mathsf{ground}(i)} \prod_{j=1}^{\lceil \log m \rceil + 2} \mathsf{PR}[b_i \notin S_j^*] \\ &\geq 1 - \sum_{b_i \in \mathsf{ground}(i)} \prod_{j=1}^{\lceil \log m \rceil + 2} 1/2 \\ &= 1 - \sum_{b_i \in \mathsf{ground}(i)} \prod_{j=1}^{\lceil \log m \rceil + 2} \mathsf{PR}[b_i \notin S_j^*] \\ &\geq 1 - \sum_{b_i \in \mathsf{ground}(i)} 1/4m \\ &= 1 - 1/4 \\ &= 3/4 \end{aligned}$$

and thus $\mathbb{E}[f(\hat{S})] \geq 3/4 f(\mathsf{ground}(f))$. On the other hand, by the linearity of expectation and the fact that f is subadditive we have:

$$\begin{aligned}\mathbb{E}[f(\hat{S})] &= \mathbb{E}[f(\bigcup S_i^*)] \\ &\leq \mathbb{E}[\sum f(S_i^*)] \\ &\leq (\lceil \log m \rceil + 2)(\sum_{S \subseteq \mathsf{ground}(f)} \alpha_S f(S))\end{aligned}$$

Therefore $\sum_{S \subseteq \mathsf{ground}(f)} \alpha_S f(S) \geq 3/4 f(\mathsf{ground}(f))/(\lceil \log m \rceil + 2)$, which means

$$\sum_{S \subseteq \mathsf{ground}(f)} \alpha_S f(S) \geq f(\mathsf{ground}(f))/(2\lceil \log m \rceil)$$

for big enough m. This shows the optimal solution of LP 12 is lower bounded by $f(\mathsf{ground}(f))/(2\lceil \log m \rceil)$ and the proof is complete. □

In what follows, based on Lemma 6.1, we provide a reduction from subadditive agents to XOS agents. An immediate corollary of Lemma 6.1 is the following:

Corollary 6.2 (of Lemma 6.1) *For any subadditive function f and integer number n, there exists an XOS function g such that*

$$g(S) \leq f(S) \qquad \forall S \subseteq \mathsf{ground}(f)$$

and

$$\mathsf{MMS}_g^n \geq \mathsf{MMS}_f^n / 2\lceil \log n \rceil.$$

Proof. By definition, we can divide the items into n disjoint sets such that the value of f for every set is at least MMS_f^n. Now, based on Lemma 6.1, we approximate f for each set with an additive function g_i wile losing a factor of at most $\lceil 2|\log \mathsf{ground}(f)| \rceil$ and finally we set $g = \max g_i$. Based on Lemma 6.1, both conditions of this lemma are satisfied by g. □

Based on Theorem 5.4 and Lemma 6.2 one can show that a $1/10\lceil \log m \rceil$-MMS allocation is always possible for subadditive agents.

Theorem 6.3 *The fair allocation problem with subadditive agents admits a $1/10\lceil \log m \rceil$-MMS allocation.*

7 Acknoledgment

We would like to thank the anonymouse reviewers for their thoughtful comments and direction.

References

[1] R. Alijani, M. Farhadi, M. Ghodsi, M. Seddighin, and A. S. Tajik. Envy-free mechanisms with minimum number of cuts. In *AAAI*, pages 312–318, 2017.

[2] G. Amanatidis, E. Markakis, A. Nikzad, and A. Saberi. Approximation algorithms for computing maximin share allocations. In *ICALP 2015*.

[3] C. Annamalai, C. Kalaitzis, and O. Svensson. Combinatorial algorithm for restricted max-min fair allocation. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1357–1372. SIAM, 2015.

[4] A. Asadpour and A. Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. In *STOC 2007*.

[5] A. Bachem and W. Kern. *Linear programming duality*. Springer, 1992.

[6] N. Bansal and M. Sviridenko. The santa claus problem. In *STOC 2006*.

[7] S. Barman and S. K. K. Murthy. Approximation algorithms for maximin fair division. In *EC 2017*.

[8] M. Bateni, M. Hajiaghayi, and M. Zadimoghaddam. Submodular secretary problem and extensions. In *APPROX-RANDOM 2010*.

[9] K. Bhawalkar and T. Roughgarden. Welfare guarantees for combinatorial auctions with item bidding. In *SODA 2011*.

[10] L. Blumrosen and S. Dobzinski. Welfare maximization in congestion games.

[11] S. Bouveret and M. Lemaître. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. In *AAMAS 2014*.

[12] S. J. Brams and A. D. Taylor. An envy-free cake division protocol. *American Mathematical Monthly*, pages 9–18, 1995.

[13] N. Buchbinder, M. Feldman, J. Naor, and R. Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *FOCS 2012*.

[14] N. Buchbinder, M. Feldman, J. Seffi, and R. Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *FOCS 2012*.

[15] E. Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

[16] I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang. The unreasonable fairness of maximum nash welfare. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 305–322. ACM, 2016.

[17] D. Chakrabarty, J. Chuzhoy, and S. Khanna. On allocating goods to maximize fairness. In *FOCS 2009*.

[18] G. Christodoulou, A. Kovács, and M. Schapira. Bayesian combinatorial auctions. In *ICALP 2008*.

[19] S. Dobzinski, N. Nisan, and M. Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *STOC 2005*.

[20] S. Dobzinski and M. Schapira. An improved approximation algorithm for combinatorial auctions with submodular bidders. In *SODA 2006*.

[21] L. E. Dubins and E. H. Spanier. How to cut a cake fairly. *American mathematical monthly*, pages 1–17, 1961.

[22] L. Epstein and A. Levin. An efficient polynomial time approximation scheme for load balancing on uniformly related machines. *Mathematical Programming*, 147(1-2):1–23, 2014.

[23] A. Farhadi, M. Hajiaghayi, M. Ghodsi, S. Lahaie, D. Pennock, M. Seddighin, S. Seddighin, and H. Yami. Fair allocation of indivisible goods to asymmetric agents. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 1535–1537. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[24] U. Feige. On allocations that maximize fairness. In *SODA 2007*.

[25] U. Feige. On maximizing welfare when utility functions are subadditive. In *STOC 2006*.

[26] U. Feige, V. S. Mirrokni, and J. Vondrak. Maximizing non-monotone submodular functions. In *FOCS 2007*.

[27] U. Feige and J. Vondrak. Approximation algorithms for allocation problems: Improving the factor of 1-1/e. In *FOCS 2006*.

[28] M. Feldman, H. Fu, N. Gravin, and B. Lucier. Simultaneous auctions are (almost) efficient. In *STOC 2013*.

[29] M. Feldman, N. Gravin, and B. Lucier. Combinatorial auctions via posted prices. In *SODA 2015*.

[30] D. K. Foley. Resource allocation and the public sector. *YALE ECON ESSAYS, VOL 7, NO 1, PP 45-98, SPRING 1967. 7 FIG, 13 REF.*, 1967.

[31] H. Fu, R. Kleinberg, and R. Lavi. Conditional equilibrium outcomes via ascending price processes with applications to combinatorial auctions with item bidding. In *ACM EC 2012*.

[32] S. Fujishige. *Submodular functions and optimization*, volume 58. Elsevier, 2005.

[33] M. X. Goemans, N. J. Harvey, S. Iwata, and V. Mirrokni. Approximating submodular functions everywhere. In *SODA 2009*.

[34] D. Golovin. Max-min fair allocation of indivisible goods. 2005.

[35] A. Gupta, A. Roth, G. Schoenebeck, and K. Talwar. Constrained non-monotone submodular maximization: Offline and secretary algorithms. In *WINE 2010*.

[36] V. Hugo. *Les misérables*. Modern library, 1862.

[37] G. Kim, E. P. Xing, L. Fei-Fei, and T. Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *ICCV 2011*.

[38] A. Krause. Sfo: A toolbox for submodular function optimization. *The Journal of Machine Learning Research*, 11:1141–1144, 2010.

[39] D. Kurokawa, A. D. Procaccia, and J. Wang. When can the maximin share guarantee be guaranteed? In *AAAI 2015*.

[40] J. Lee, V. S. Mirrokni, V. Nagarajan, and M. Sviridenko. Non-monotone submodular maximization under matroid and knapsack constraints. In *STOC 2009*.

[41] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. In *ACM EC 2001*.

[42] R. P. Leme. Gross substitutability: An algorithmic survey. *preprint*, 2014.

[43] R. J. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *ACM EC 2004*.

[44] I. Milchtaich. Congestion games with player-specific payoff functions. In *MFCS 2007*.

[45] M. Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, pages 234–243. Springer, 1978.

[46] A. D. Procaccia and J. Wang. Fair enough: Guaranteeing approximate maximin shares. In *ACM EC 2014*.

[47] H. Steinhaus. The problem of fair division. *Econometrica*, 16(1), 1948.

[48] W. Suksompong. Approximate maximin shares for groups of agents. *arXiv preprint arXiv:1706.09869*, 2017.

[49] V. Syrgkanis. Bayesian games and the smoothness framework. *arXiv preprint arXiv:1203.5155*, 2012.

[50] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC 2008*.

[51] D. B. West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

A A 4/5-MMS Allocation for Four Agents

In this section we propose an algorithm to find a 4/5-MMS for $n = 4$ in the additive setting. Since the number of agents is exactly 4, we assume $\mathcal{N} = \{a_1, a_2, a_3, a_4\}$. Again, for simplicity we assume $\mathsf{MMS}_i = 1$ for every $a_i \in \mathcal{N}$. In general, our algorithm is consisted of three main steps: first, a_1 optimally partitions the items into 4 bundles with values at least 1 to him. Next, a_2 selects three of the bundles and repartitions them. Finally, we satisfy one of a_1 or a_2 with a bundle and solve the problem for remaining agents and items via Lemma A.4. Without loss of generality, we assume the valuation of every agent a_i for every bundle in his optimal n-partitioning is exactly equal to 1. Therefore, from here on, we assume that the summation of the values of the items within the same bundle for every agent is at most 1. In addition, we suppose that the problem is 4/5-irreducible, since Lemma 2.1 narrows down the problem into such instances. Thus, by Lemma 3.1, the value of every item is less than 4/5 to any agent. We begin this section by stating a number of definitions and observations. In this section, we use the term *bundle* to refer to a set of items.

Definition A.1 *A set S of bundles is perfect for a set T of agents, if (i) $|S| = |T|$ and (ii) there exists an allocation of the bundles in S to the agents of T such that all the agents in T are satisfied by their allocated bundle.*

Observation A.1 *Let a_i be an agent and S be a set of items where $V_i(\{b_j\}) \leq v$ for every item $b_j \in S$. If $V_i(S) > v$, then there exists a subset $S' \subseteq S$ of items such that $v \leq V_i(S') < 2v$.*

Proof. We begin with an empty set S' and add the items of S to S' one by one, until $V_i(S')$ exceeds v. Before adding the last element to S', the valuation of a_i for S' was no more than v and every item alone is of value less than v to a_i. Therefore, after adding the last item to S, its value is less than $2v$ to a_i. □

Definition A.2 *For a bundle B of items that satisfies a_i, the core of B with respect to agent a_i, denoted by $C_i(B)$, is defined as follows: let $m_1, m_2, .., m_k$ be the items of B in the increasing order of their values to a_i. Then $C_i(B) = \{m_j, m_{j+1}, ..., m_k\}$, where j is the highest index, such that set of items $\{m_j, m_{j+1}, ..., m_k\}$ satisfies a_i.*

Note that for every subset B with $V_i(B) \geq 4/5$, $C_i(B)$ is a subset of B with the minimum size that satisfies a_i. Since the items in $C_i(B)$ satisfy a_i, we have $V_i(C_i(B)) \geq 4/5$. On the other hand, by the fact that $|C_i(B)|$ is minimal, removing any item from $C_i(B)$ results in a subset that no longer satisfies a_i. Thus, Observation A.2 holds.

Observation A.2 *If $V_i(C_i(B)) = 4/5 + \beta$, then the value of every item in $C_i(B)$ is more than β for a_i.*

By the fact the value of every item in B is less than 4/5 we have Observation A.3.

Observation A.3 *For every agent a_i and any subset B of items, $V_i(C_i(B)) < 8/5$.*

Lemma A.3 *Suppose that $S = \{X, Y, Z\}$ is a 3-partitioning of a set of items with the following properties for an agent a_i:*

(i). $V_i(X) < 4/5$ and $V_i(Y) < 4/5$.

(ii). $V_i(X \cup Y \cup Z) > 16/5$.

Then we can move some items from Z to an arbitrary bundle of $\{X, Y\}$, such that, both Z and the corresponding bundle will be worth at least 4/5 to a_i.

Proof. Since $V_i(X \cup Y \cup Z) > 16/5$, $V_i(Z) > 8/5$ holds. Moreover, by Observation A.3, we have $V_i(C_i(Z)) \leq 8/5$. Considering $Z' = Z \setminus C_i(Z)$, we have

$$V_i(X \cup Y \cup Z') \geq 8/5.$$

According to the fact that $V_i(X) < 4/8$ and $V_i(Y) < 4/5$, we have

$$V_i(X \cup Z') \geq 4/5$$

and

$$V_i(Y \cup Z') \geq 4/5.$$

□

Lemma A.4 *Let $S = \{X, Y, Z\}$ be a set of three bundles of items, such that*

(i). $V_1(X) \geq 4/5, V_1(Y) \geq 4/5, V_1(Z) \geq 4/5$.

(ii). $V_2(X \cup Y \cup Z) > 16/5$.

(iii). $V_3(X \cup Y \cup Z) \geq 3$.

Then a $4/5$-MMS *allocation of $X \cup Y \cup Z$ to the agents a_1, a_2, a_3 is possible.*

Proof. If a_2 can be satisfied with two different bundles, then trivially S is perfect. Otherwise, a_2 is satisfied with only one bundle, say Z. By Lemma A.3, a_2 can transfer some items from Z to Y, such that both bundles satisfy him. After moving the items, both Y and Z satisfy a_2, and bundle X and Y, satisfy a_1. One the other hand, since $V_3(X \cup Y \cup Z) \geq 3$, a_3 is satisfied with at least one bundle. Its easy to observe that for any valuation of bundles X, Y, Z for a_3, the set of bundles is perfect. □

Lemma A.5 *Let $S = \{X, Y, Z, T\}$ be a 4-partitioning of $\mathcal{M}$ and a_i be an arbitrary agent. Then a_i can select 3 bundles and re-partition them into three new bundles in such a way that each bundle will be worth at least $4/5$ to a_i.*

Proof. Consider bundles X, Y, Z, T. If more than two of them satisfy a_i, then the selection is trivial. Furthermore, if only one bundle satisfies a_i, then by Lemma A.3, we can move some items from the satisfying bundle to another bundle, such that both bundles satisfy a_i. Thus, without loss of generality, we assume that bundles Z and T satisfy a_i.

Let $Z' = Z \setminus C_i(Z)$ and $T' = T \setminus C_i(T)$. Without loss of generality, we assume $V_i(X) \geq V_i(Y)$ and let $X' = X \cup Z' \cup T'$. If $V_i(X') \geq 4/5$, then the proof is trivial. Thus, suppose that $V_i(X') < 4/5$.

Consider the value of bundles as

$$V_i(X) = 4/5 - \epsilon_1,$$

$$V_i(Y) = 4/5 - \epsilon_2,$$

$$V_i(C_i(Z)) = 6/5 + \epsilon_3,$$

and

$$V_i(C_i(T)) = 6/5 + \epsilon_4$$

where $\epsilon_1 \leq \epsilon_2$ and $\epsilon_3 \leq \epsilon 4$. Note that ϵ_3 can be negative. By the fact that total value of the items equals 4 for all of the agents, we assume that

$$\epsilon_1 + \epsilon_2 = \epsilon_3 + \epsilon_4$$

and hence

$$\epsilon_1 \leq \epsilon_4.$$

Now, we explore the properties of the items in $C_i(T)$. Regarding Observation A.2, every item in $C_i(T)$ is worth more than $2/5+\epsilon_4$ to a_i. Hence, $C_i(T)$ cannot contain more than 2 items, since value of every pair of items in $C_i(T)$ is more than $4/5$. Moreover, $C_i(T)$ cannot contain one item and hence, $C_i(T)$ contains exactly two items. Let b_1 and b_2 be these two items. Since $V_i(T) = 6/5 + \epsilon_4$, at least one of these two items, say b_2 is worth at least $3/5 + \epsilon_4/2$ to a_i. Thus, in summary, $C_i(T)$ contains two items b_1 and b_2 with

$$V_i(\{b_1\}) > 2/5 + \epsilon_4,$$
$$V_i(\{b_2\}) \geq 3/5 + \epsilon_4/2.$$

Next, we characterise the items in X'. For Bundle X', let B be the set of items with a value less than $1/5 - \epsilon_4/2$. If $V_i(B) \geq 1/5 - \epsilon_4/2$, then Observation A.1 states that there exists a subset B' of B, such that:

$$1/5 - \epsilon_4/2 \leq V_i(B') < 2/5 - \epsilon_4.$$

Therefore, Bundles $B' \cup \{b_2\}$ and $(X' \setminus B') \cup \{b_1\}$ satisfy a_i. These two bundles together with $C_i(Z)$ result in three bundles that satisfy a_i. Thus, $V_i(B) < 1/5 - \epsilon_4/2$.

Finally, regarding the fact that $V_i(B) < 1/5 - \epsilon_4/2$, we have

$$V_i(X' \setminus B) > 3/5 - \epsilon_1 + \epsilon_4/2$$

For this case, we show that $X' \setminus B$ contains exactly one item. Otherwise, at least one of these items, say b_3, is worth less than $3/10 - \epsilon_1/2 + \epsilon_4/4$ and therefore, for the bundles $\{b_3\} \cup \{b_2\}$ and $(X' \setminus \{b_3\}) \cup \{b_1\}$ we have:

$$V_i(\{b_3\} \cup \{b_2\}) \geq 1/5 - \epsilon_4/2 + 3/5 + \epsilon_4/2 \geq 4/5$$

and

$$V_i((X' \setminus \{b_3\}) \cup \{b_1\}) \geq 4/5 - \epsilon_1 - 3/10 + \epsilon_1/2 - \epsilon_4/4 + 2/5 + \epsilon_4 \geq 4/5$$

respectively. These two bundles along with $C_i(Z)$ form 3 bundles that satisfy a_i. Therefore, we conclude that $X' \setminus B$ contains an item b_3 with a value more than $3/5 - \epsilon_1 + \epsilon_4/2$ to a_i. The rest of the items in X' belong to B that are in total worth less than $1/5 - \epsilon_4/2$.

Note that $X \subseteq X'$. Therefore, consider the 4−partitioning of a_i and remove the bundle containing b_3. Also, remove the items with value less than $1/5 - \epsilon_4/2$ to a_i in X, from their corresponding bundles. Three bundles with value of each to a_i more than

$$1 - 1/5 + \epsilon_4/2 \geq 4/5,$$

with all of their items from Y, Z and T remain. Thus, a_i can make three satisfying bundles with items in Y, Z, T. □ Based on what we showed so far, we prove Theorem A.6.

Theorem A.6 *A* 4/5-MMS *allocation for* $n = 4$ *is possible in the additive setting.*

Proof. Consider the optimal 4-partitioning of $\mathcal{M}$ with respect to a_1. Now, ask a_2 to select 3 bundles and re-partition them, such that he can be satisfied with all the three bundles. Based on Lemma A.5, such a repartitioning is always possible. Due to the Pigeonhole principle, at least one of these three bundles still satisfies a_1. Let $S = \{X, Y, Z, T\}$ be the resulting bundles and without loss of generality, suppose that bundles X, Y satisfy a_1 and bundles Y, Z, T satisfy a_2.

Now, consider agents a_3 and a_4 and let ϕ be the set of bundles that satisfy a_3 or a_4. There are only two cases, in which S is not perfect (recall definition A.1):

(i). $\phi \subseteq \{X, Y\}$: a_1, selects three bundles X, Y and one of Z or T, say Z and re-partitions them to three satisfying bundles. Now, give bundle T to a_2. According to Lemma A.4, items of X, Y, Z can satisfy the remaining three agents.

(ii). $|\phi| = 1, \phi \notin \{X, Y\}$: give X to a_1 and allocate the items of $Y \cup Z \cup T$ to a_2, a_3, a_4, using Lemma A.4.

□

B Omitted Proofs of Section 3.2

Proof of Lemma 3.1: The key idea is that given $\mathsf{MMS}_i \geq 1$ for an agent a_i, then for every item $b_j \in \mathcal{M}$ we have $\mathsf{MMS}_i^{n-1}(\mathcal{M} \setminus b_j) \geq 1$. This holds since removing an item from $\mathcal{M}$ will diminish the value of at most one partition in the optimal n partitioning of the items. Therefore, at least $n-1$ partitions have a value of 1 or more to a_i and thus $\mathsf{MMS}_i^{n-1}(\mathcal{M} \setminus b_j) \geq 1$. The rest of the proof follows from the definition of α-irreducibility. If the valuation of an item b_j to an agent a_i is at least α, then the problem is α-reducible since if we allocate b_j to a_i, we have

$$\mathsf{MMS}_{V_k}^{n-1}(\mathcal{M} \setminus \{b_j\}) \geq 1$$

for every agent $a_k \neq a_i$. This contradicts with the α-irreducibility assumption. □

Proof of Lemma 3.2: Suppose for the sake of contradiction that for every agent $a_{i'} \neq a_i$ we have $V_{i'}(\{b_j, b_k\}) \leq 1$. By this assumption, we show

$$\mathsf{MMS}_{i'}^{n-1}(\mathcal{M} \setminus \{b_j, b_k\}) \geq 1 \tag{16}$$

holds. This is true since removing two items b_j and b_k from $\mathcal{M}$ decreases the value of at most two partitions of the optimal partitioning of $\mathcal{M}$ for $\mathsf{MMS}_{i'}$. If $n-1$ partitions remain intact, then Inequality (16) trivially holds. If not, merging the two partitions that initially contained b_j and b_k results in a partition with value at least 1 to a_i. This partition together with the $n-2$ remaining partitions result in a desirable partitioning of $\mathcal{M}$ into $n-1$ partitions. Therefore, Inequality (16) holds for any agent $a_{i'}$, and this implies that by allocating $S = \{b_j, b_k\}$ to a_i, not only does $V_i(S) \geq 3/4$ hold, but also for every $a_{i'} \neq a_i$ we have

$$\mathsf{MMS}_{i'}^{n-1}(\mathcal{M} \setminus \{b_j, b_k\}) \geq 1$$

which means the problem is 3/4-reducible, and it contradicts our assumption. □

Proof of Lemma 3.3: The proof for this lemma is obtained by applying Lemma 3.2, $|T|$ times. Consider an agent $a_i \notin T$. According to the argument in Lemma 3.2, if we assign b_{j_1} and b_{j_2} to a_{i_1}, a_i can partition the items in $\mathcal{M} \setminus \{b_{j_1}\ b_{j_2}\}$ into $n-1$ partitions with value at least 1 to a_i, i.e.

$$\mathsf{MMS}_i^{n-1}(\mathcal{M} \setminus \{b_{j_1}, b_{j_2}\}) \geq 1.$$

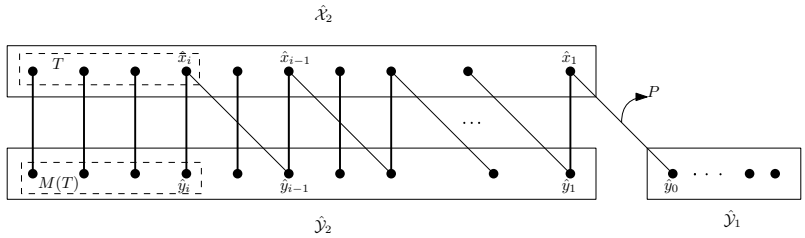

Figure 11: Alternating path P connects $\hat{\mathcal{Y}}_1$ to $\hat{\mathcal{Y}}_2$ and intersects T

By the same deduction, after assigning b_{j_3} and b_{j_4} to a_{i_2}, we have

$$\mathsf{MMS}_i^{n-2}(\mathcal{M} \setminus \{b_{j_1}, b_{j_2}, b_{j_3}, b_{j_4}\}) \geq 1.$$

By repeating above argument $|T|$ times, we have:

$$\mathsf{MMS}_i^{n-|T|}(\mathcal{M} \setminus S) \geq 1.$$

On the other hand, by condition (II), every agent a_{i_k} satisfies with items $b_{j_{2k-1}}$ and $b_{j_{2k}}$. This means that we can reduce the instance by satisfying the agents in T by the items in S, which is a contradiction by the irreducibility assumption.

□

Proof of Lemma 3.6: We define $\hat{\mathcal{Y}}_1$ as the set of vertices in $\hat{\mathcal{Y}}$ that are not saturated by M, and $\hat{\mathcal{Y}}_2$ as the set of vertices in $\hat{\mathcal{Y}}$ that are connected to $\hat{\mathcal{Y}}_1$ by an alternating path. Moreover, let $\hat{\mathcal{X}}_2 = M(\hat{\mathcal{Y}}_2)$. By definition, $F_H(M, \hat{\mathcal{X}}) = \hat{\mathcal{X}} \setminus \hat{\mathcal{X}}_2$ (See Figure 8). As discussed before, all the vertices in $\hat{\mathcal{X}}_2$ are saturated by M. Consequently, all the vertices of T are saturated by M and $|N(T)| \geq |T|$.

Let $M(T)$ be the set of vertices which are matched to the vertices of T in M. We know that every vertex of T is present in at least one of the alternating paths which connect $\hat{\mathcal{Y}}_1$ to $\hat{\mathcal{Y}}_2$. Let

$$P = \langle \hat{y}_0, \hat{x}_1, \hat{y}_1, \hat{x}_2, \hat{y}_2, \ldots, \hat{x}_k, \hat{y}_k \rangle$$

be one of these paths that includes at least one of the vertices of T. Since P is an alternating path which connects $\hat{\mathcal{Y}}_1$ to $\hat{\mathcal{Y}}_2$, $\hat{y}_0 \in \hat{\mathcal{Y}}_1$ (see Figure 11). In addition, according to the definition of alternating path, every edge $(\hat{x}_j, \hat{y}_j)$ of P belongs to M and every edge $(\hat{x}_j, \hat{y}_{j-1})$ does not belong to M.

Let $\hat{x}_i$ be the first vertex of T that appears in P. We know that the edge $(\hat{x}_i, \hat{y}_{i-1})$ does not belong to M. On the other hand, since $\hat{x}_i$ is the first vertex of T in M, $\hat{x}_{i-1} \notin T$. Note that $\hat{y}_{i-1}$ does not belong to $M(T)$, since every vertex of $M(T)$ is matched with a vertex of T in M and $(\hat{x}_{i-1}, \hat{y}_{i-1})$ is in M. The fact that $\hat{y}_{i-1} \notin M(T)$ means $N(T)$ contains at least one vertex that is not in $M(T)$. Since all the vertices in $M(T)$ are in $N(T)$, $|N(T)| > |M(T)|$ and hence, $|N(T)| > |T|$.

□

Proof of Lemma 3.7: If $F_H(M, \hat{\mathcal{X}}) = \emptyset$, according to Lemma 3.6,

$$\forall T \subseteq \hat{\mathcal{X}} \qquad |N(T)| > |T|.$$

On the other hand, suppose that for all $T \subseteq \hat{\mathcal{X}}$ we have $|N(T)| > |T|$. We show that $F_H(M, \hat{\mathcal{X}}) = \emptyset$. For the sake of contradiction, assume that $F_H(M, \hat{\mathcal{X}}) \neq \emptyset$ and let $T = F_H(M, \hat{\mathcal{X}})$. Since there exists a matching from T to $N(T)$ that saturates all the vertices of $N(T)$, we have $|T| \geq |N(T)|$, which is a contradiction. Hence, $F_H(M, \hat{\mathcal{X}}) = \emptyset$. □

Proof of Lemma 3.11: Consider a cycle L in G_C. For each vertex $v_j \in L$, there is at least one vertex $v_i \in L$ such that a_i envies a_j. Therefore, Considering S as the set of agents with vertices in L, none of the agents of S is a loser. By the same deduction, none of the agents of S is a winner. But this contradicts the fact that the set C is cycle-envy-free. □

Proof of Lemma 3.13: We describe the proof for the first condition in more details. The proof for the second condition is almost the same as the first condition.

The first condition: Suppose that there exists no such vertex. Our goal is to find a new matching of H with the same cardinality, but with more weight. To this end, we construct a directed graph H' from H as follows: for each $\hat{y}_j \in T$ we consider a vertex v_j in $V(H')$. Furthermore, there is a directed edge from v_j to v_i in H', if and only if $w(\hat{x}_j, \hat{y}_j) < w(\hat{x}_i, \hat{y}_j)$ in H.

If there exists a vertex v_j with out-degree zero in H', then $\hat{y}_j$ is the desired winner in T, since

$$\forall \hat{y}_i \in H, w(\hat{x}_j, \hat{y}_j) \geq w(\hat{x}_i, \hat{y}_j).$$

Otherwise, the out-degree of every vertex in T is non-zero. Therefore, H' has at least one cycle $L = \langle v_{l_1}, v_{l_2}, \ldots, v_{l_{|L|}} \rangle$. Now, if we change matching M by removing the set of edges

$$\{(\hat{y}_{l_1}, \hat{x}_{l_1}), (\hat{y}_{l_2}, \hat{x}_{l_2}), \ldots, (\hat{y}_{l_{|L|}}, \hat{x}_{l_{|L|}})\}$$

from M and adding

$$\{(\hat{y}_{l_1}, \hat{x}_{l_2}), (\hat{y}_{l_2}, \hat{x}_{l_3}), \ldots, (\hat{y}_{l_{|L|}}, \hat{x}_{l_1})\}$$

to M, the weight of our matching will be increased. Note that by the definition of an edge in H', we have

$$w(\hat{x}_{l_2}, \hat{y}_{l_1}) > w(\hat{x}_{l_1}, \hat{y}_{l_1}), w(\hat{x}_{l_3}, \hat{y}_{l_2}) > w(\hat{x}_{l_2}, \hat{y}_{l_2}), \ldots, w(\hat{x}_{l_1}, \hat{y}_{l_{|L|}}) > w(\hat{x}_{l_{|L|}}, \hat{y}_{l_{|L|}}).$$

But this contradicts the fact that M was MCMWM of H.

The second condition: Similar to the proof of the first condition, we construct a new directed graph H' from H where we have a vertex v_j in H' for each vertex $\hat{y}_j$ in T. For every pair $\hat{y}_i$ and $\hat{y}_j$ which are members of T we connect v_i to v_j with a directed edge in H' if

$$w(\hat{x}_j, \hat{y}_i) > w(\hat{x}_i, \hat{y}_i)$$

in H and $(\hat{x}_j, \hat{y}_i) \in E(H)$. Note that if H' contains a vertex v_i with in-degree equal to zero, then $\hat{y}_i$ is the desired loser in T. Thus, suppose that no vertex in H' has in-degree zero and hence, H' has a directed cycle. Let $L = \langle \hat{y}_{l_1}, \hat{y}_{l_2}, \ldots, \hat{y}_{l_{|L|}} \rangle$ be a directed cycle in H'. Similar to the proof of the previous condition, we leverage L to alter M to a new matching with more weight, which is a contradiction by the maximality of M.

The third condition: If $w(\hat{x}_i, \hat{y}_i) < w(\hat{x}_j, \hat{y}_i)$, we can replace the edge between $\hat{x}_i$ and $\hat{y}_i$ by $(\hat{x}_j, \hat{y}_i)$ in M which yields a matching with a greater weight. This contradicts the maximality of M. □

C Omitted Proofs of Section 3.3

Proof of Lemma 3.14: By definition, there is no edge between the vertices of $F_{G_{1/2}}(M, \mathcal{X}_{1/2})$ and $\mathcal{Y}_{1/2} \setminus N(F_{G_{1/2}}(M, \mathcal{X}_{1/2}))$ in $G_{1/2}$. Furthermore, all the items are in worth less than $1/2$ for the agents corresponding to the vertices in $\mathcal{Y} \setminus \mathcal{Y}_{1/2}$. Thus, for every agent a_i and every item b_j with

$y_i \in \mathcal{Y} \setminus N(F_{G_{1/2}}(M, \mathcal{X}_{1/2}))$ and $x_j \in F_{G_{1/2}}(M, \mathcal{X}_{1/2})$, we have $V_i(b_j) < 1/2$. According to the fact that the agents that are not selected in the clustering of $\mathcal{C}_1$ either belong to $\mathcal{C}_2$ or $\mathcal{C}_3$, we have:

$$\forall a_j \in \mathcal{C}_1 \qquad V_i(f_j) < 1/2.$$

□

Proof of Lemma 3.15: First, we prove Lemma C.1. This lemma ensures that there exists a matching in G_1 that saturates all the vertices in W_1. Lemma C.1 is a consequence of irreducibility. In fact, we show that if the condition in Lemma C.1 does not hold, the instance is reducible.

Lemma C.1 *For graph G_1, we have*

$$\forall R \subseteq W_1, \qquad |N(R)| > |R|.$$

Proof. Let M_1 a matching with the maximum number of edges in G_1 . Regarding Lemma 3.7, it only suffices to show that $F_{G_1}(M_1, W_1)$ is empty. For the sake of contradiction, suppose that $F_{G_1}(M_1, W_1)$ is not empty. As mentioned before, there exists a matching between $F_{G_1}(M_1, W_1)$ and $N(F_{G_1}(M_1, W_1))$ that saturates all the vertices in $N(F_{G_1}(M_1, W_1))$. Let

$$M_S = \{(x_{j_1}, y_{i_1}), (x_{j_2}, y_{i_2}), \ldots, (x_{j_k}, y_{i_k})\}$$

be this matching. We show that the set of agents

$$T = \{a_{i_1}, a_{i_2}, \ldots, a_{i_k}\}$$

and the set of items

$$S = \{f_{i_1}, b_{j_1}, f_{i_2}, b_{j_2}, \ldots, f_{i_k}, b_{j_k}\}.$$

have all three conditions in Lemma 3.3 (Note that f_{i_l} contains exactly one item). The first condition is trivial: $|S| = 2|T|$. Regarding the definition of an edge in G_1, we know that $f_{i_l} \cup \{b_{j_l}\}$ satisfy a_{i_l} and hence, the second condition is held as well. For the third condition, we should prove that for every agent a_{i_l} in T,

$$V_{i'}(f_{i_l} \cup \{b_{j_l}\}) < 1 \qquad \forall a_{i'} \notin T.$$

To show this, we consider two cases separately. First, if $a_{i'} \notin C_1$, by Lemma 3.14, $V_{i'}(f_{i_l}) < 1/2$ and by Observation 3.2, $V_{i'}(\{b_{j_l}\}) < 1/2$, which means $V_{i'}(f_{i_l} \cup \{b_{j_l}\}) < 1$.

Moreover, consider the case that $a_{i'} \in C_1$. Note that since $a_{i'} \notin T$, it's corresponding vertex $y_{i'}$ is not in $N(F_{G_1}(M_1, W_1))$, which means:

$$y_{i'} \in V_{C_1} \setminus N(F_{G_1}(M_1, W_1)).$$

By the definition of $N(F_{G_1}(M_1, W_1)$, there is no edge between $y_{i'}$ and x_{j_l} and hence, $V_{i'}(\{b_{j_l}\}) < \epsilon_{i'} \leq 1/4$. On the other hand, by the irreducibility assumption and the fact that f_{i_l} contains exactly one item, $V_{i'}(f_{i_l}) < 3/4$. Thus, $V_{i'}(f_{i_l} \cup \{b_{j_l}\}) < 1$.

As a result, $V_{i'}(f_{i_l} \cup \{b_{j_l}\}) < 1$ for every agent $a_{i'} \notin T$ which means the third condition of Lemma 3.3 is held as well. Thus, regarding Lemma 3.3, the instance is reducible. But this contradicts the irreducibility assumption. □

The rest of the proof of Lemma 3.15 is as follows. Since we used MCMWM to build cluster $\mathcal{C}_1$, regarding Lemma 3.13, $\mathcal{C}_1$ is cycle-envy-free. Consider the topological ordering of $\mathcal{C}_1$ and let p_{a_i} be the position of a_i in this ordering. More precisely, $p_{a_i} = k$ if a_i is the k-th agent in the topological ordering of $\mathcal{C}_1$.

According to Lemma C.1, the condition of Halls Theorem holds for graph G_1 and as a result there exists a matching in G_1 that saturates all the vertices in W_1. Among all possible maximum matchings of G_1, let M_1 be a maximum matching that minimizes

$$p_{M_1} = \sum_{y_i \in M_1} p_{a_i}.$$

We claim that M_1 is the desired matching described in Lemma 3.15. To prove our claim, we must show that for any edge $(x_i, y_j) \in M_1$ and any unsaturated vertex $y_k \in N(x_i)$, a_j is a loser for the set $\{a_j, a_k\}$, which means a_k does not envy a_j. Note that if a_k envies a_j, a_k appears before a_j in the topological ordering of $\mathcal{C}_1$ which means $p_{a_k} < p_{a_j}$. Therefore, if we replace (x_i, y_j) by (x_i, y_k) in M_1, p_{M_1} will be decreased that contradicts the minimality of p_{M_1}. □

Proof of Lemma 3.16: Let b_k be the item assigned to a_j in the refinement of $\mathcal{C}_1$. Since $x_k \in W_1$, according to Observation 3.2, $V_i(g_j) < 1/2$. □

Proof of Lemma 3.17: Let a_j be an agent in $\mathcal{S}_1^r$. First, note that $|f_j| = |g_j| = 1$. Lemma 3.14 together with Observation 3.2 state that $V_i(f_j \cup g_j) < 1$. According to Inequality (16), we have

$$\mathsf{MMS}_{V_i}^{|\mathcal{N}\setminus a_j|}(\mathcal{M} \setminus f_j \cup g_j) \geq 1. \tag{17}$$

Note that Equation (17) holds for every agent in $\mathcal{S}_1^r$. Applying Equation (17) to all the agents of $\mathcal{S}_1^r$ yields

$$\mathsf{MMS}_{V_i}^{|\mathcal{N}\setminus \mathcal{S}_1^r|}(\mathcal{M} \setminus \bigcup_{y_i \in \mathcal{S}_1^r} f_i \cup g_i) \geq 1.$$

□

Proof of Lemma 3.19: According to Observation 3.3, for any agent $a_k \in \mathcal{C}_1$ and for every $x_j \in \mathcal{X}' \setminus \mathcal{X}'_{1/2}$ we have $V_k(\{b_j\}) < \epsilon_k$. By additivity assumption, for any $a_k \in \mathcal{C}_1$ we have

$$\forall x_i, x_j \in \mathcal{X}' \setminus \mathcal{X}'_{1/2} \qquad V_k(\{b_i, b_j\}) < 2\epsilon_k.$$

□

Proof of Lemma 3.20: Suppose for the sake of contradiction that the problem is 3/4-irreducible, and there exists a vertex $y_k \in \mathcal{Y}$ such that $V_k(\{b_i, b_j\}) \geq 3/4$. According to Lemma 3.2 there exists an agent $a_{k'} \neq a_k$ such that

$$V_{k'}(\{b_i, b_j\}) \geq 1.$$

Since the valuations are additive, we know that one of the inequalities $V_{k'}(\{b_i\}) \geq 1/2$ or $V_{k'}(\{b_j\}) \geq 1/2$ are held, which is contradiction, since we know both x_i and x_j belong to $\mathcal{X}' \setminus \mathcal{X}'_{1/2}$. □

Proof of Lemma 3.22: We prove Lemma 3.22 in two steps. Firstly, we show that

$$|F_{G'_{1/2}}(M', \mathcal{X}'_{1/2})| \leq |N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))|. \tag{18}$$

Furthermore, we prove

$$|F_{G'_{1/2}}(M', \mathcal{X}'_{1/2})| \geq |N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))|. \tag{19}$$

Inequalities (19) and (19) yields

$$|F_{G'_{1/2}}(M', \mathcal{X}'_{1/2})| = |N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))|. \tag{20}$$

To show Inequality (18)**,** argue that before Algorithm 1 starts, we have

$$F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}) = \emptyset$$

and

$$N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2})) = \emptyset$$

and all the vertices in $\mathcal{X}'_{1/2}$ are saturated by M'. In each step of Algorithm 1, we add a new vertex to $\mathcal{X}'_{1/2}$, and the size of the maximum matching M' is increased by one. Therefore, after each step of Algorithm 1, all of the vertices in $\mathcal{X}'_{1/2}$ remain saturated by M'. Since $F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}) \subseteq \mathcal{X}'_{1/2}$, all the vertices of $F_{G'_{1/2}}(M', \mathcal{X}'_{1/2})$ are also saturated by M', which means

$$|F_{G'_{1/2}}(M', \mathcal{X}'_{1/2})| \leq |N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))|.$$

To prove Inequality (19), note that by definition, $F_{G'_{1/2}}(M', \mathcal{X}'_{1/2})$ has a property that there exists a matching from $F_{G'_{1/2}}(M', \mathcal{X}'_{1/2})$ to $N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))$ that saturates all the vertices of $N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))$. Therefore, we have

$$|F_{G'_{1/2}}(M', \mathcal{X}'_{1/2})| \geq |N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))|.$$

This completes the proof. □

Proof of Lemma 3.23: Firstly, we clarify what agents are in $\mathcal{C}_3$. Roughly speaking, the agents that are not selected for Clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ are in $\mathcal{C}_3$. Thus, the agents in $\mathcal{C}_3$ correspond to the vertices in

$$\mathcal{Y}' \setminus N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))$$
$$= \big(\mathcal{Y}' \setminus \mathcal{Y}'_{1/2}\big) \cup \big(\mathcal{Y}'_{1/2} \setminus N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))\big).$$

The term $\mathcal{Y}' \setminus \mathcal{Y}'_{1/2}$ refers to the vertices that are filtered in $G'_{1/2}$ which means no edge with weight at least $1/2$ is incident to any of these vertices. On the other hand, for every agent $a_j \in \mathcal{C}_2$, f_j corresponds to a vertex in $F_{G'_{1/2}}(M', \mathcal{X}'_{1/2})$. Hence, for every agent $a_j \in \mathcal{C}_2$ and every agent a_i with corresponding vertex in $\mathcal{Y}' \setminus \mathcal{Y}'_{1/2}$ we have $V_i(f_j) < 1/2$

Next, consider the term $\mathcal{Y}'_{1/2} \setminus N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))$**.** By definition, the vertices of $F_{G'_{1/2}}(M', \mathcal{X}'_{1/2})$ are only incident to the vertices of $N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))$ in $G'_{1/2}$. Regarding the definition of an edge in $G'_{1/2}$, for every agent $a_j \in \mathcal{C}_2$ and agent a_i with $y_i \in \mathcal{Y}'_{1/2} \setminus N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))$ we have $V_i(f_j) < 1/2$.

Therefore, for all $a_i \in \mathcal{C}_3$ we have:

$$\forall a_j \in \mathcal{C}_2 \qquad V_i(f_j) < 1/2.$$

□

Proof of Lemma 3.24: Regarding Observation 3.3, after refinement of $\mathcal{C}_1$, all the items with vertex in $\mathcal{X}' \setminus \mathcal{X}'_{1/2}$ are in worth less than ϵ_j for every agent $a_j \in \mathcal{C}_1$. Furthermore, note that for every agent $a_i \in \mathcal{S}_2^r$, g_i is a single item with vertex in $\mathcal{X}' \setminus \mathcal{X}'_{1/2}$. Thus, $V_j(g_i) < \epsilon_j$ for every agent $a_j \in \mathcal{C}_1$. □

Proof of Lemma 3.25: According to Algorithm 2, for any agent $a_i \in \mathcal{S}_2^r$, the corresponding vertex of the only member of g_i is in $\mathcal{X}' \setminus \mathcal{X}'_{1/2}$. Therefore, for any agent $a_j \notin \mathcal{C}_1 \cup \mathcal{C}_2$ we have $V_j(g_i) < 1/2$. Finally, note that the remaining agents that are not in $\mathcal{C}_1$ and $\mathcal{C}_2$ belong to $\mathcal{C}_3$. □

Proof of Lemma 3.26: The algorithm 1 terminates when there is no desirable pair for the agents in $T = \mathcal{Y}' \setminus N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2}))$. Furthermore, by definition, for every agent $a_i \in \mathcal{C}_3$ we have

$$y_i \in \mathcal{Y}' \setminus N(F_{G'_{1/2}}(M', \mathcal{X}'_{1/2})).$$

But at the end of Algorithm 1, no pair of vertices is desirable for a_i which means for every $x_j, x_k \in \mathcal{X}'' \setminus \mathcal{X}''_{1/2}$, we have $V_i(\{b_j, b_k\}) < 1/2$ (note that $\mathcal{X}'' \setminus \mathcal{X}''_{1/2} \subseteq \mathcal{X}' \setminus \mathcal{X}'_{1/2}$). □

D Omitted Proofs of Section 3.4

Proof of Lemma 3.28: At this point, for every agent $a_i \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3^s$, $|f_j| \leq 2$. If $|f_i| = 1$ holds, then according to Lemma 3.1, value of the item in f_i is less than 3/4 to all other agents. Moreover, if $|f_i| = 2$, then f_i corresponds to a merged vertex. In this case, by Lemmas 3.19 and 3.20, value of f_i is less than 3/4 to all other agents. □

Proof of Lemma 3.32: According to Lemma 3.26, value of every pair of items in $\mathcal{F}$ is less than 1/2 to a_i. Therefore, f_i contains at least three items. Let b_k be an arbitrary item in f_i. Since $|f_i| \geq 3$, $f_i \setminus \{b_k\}$ is non-empty. On the other hand, S is minimal and hence, none of the sets $f_i \setminus b_k$ and $\{b_k\}$ is feasible for any agent. According to the definition of feasibility for the agents of $\mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3^s \cup \mathcal{C}_3^b$, we have

$$\forall a_j \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3^s \cup \mathcal{C}_3^b \qquad V_j(f_i \setminus \{b_k\}) < \epsilon_j$$

and

$$\forall a_j \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3^s \cup \mathcal{C}_3^b \qquad V_j(\{b_k\}) < \epsilon_j$$

which means

$$\forall a_j \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3^s \cup \mathcal{C}_3^b \qquad V_j(f_i) < 2\epsilon_j.$$

□

Proof of Lemma 3.33: The Lemma trivially holds for $\mathcal{C}_1$ and $\mathcal{C}_2$, since removing an agent from a cycle-envy-free set preserves this property. For $\mathcal{C}_3^s$, there may be multiple rounds that an agent is added to $\mathcal{C}_3^s$. We show that adding an agent to $\mathcal{C}_3^s$ preserves cycle-envy-freeness as well.

For the sake of contradiction, let $\mathbb{R}_z$ be the first round in which adding an agent a_i to $\mathcal{C}_3^s$ results in a set, that is no longer cycle-envy-free. Since $\mathcal{C}_3^s \setminus \{a_i\}$ is cycle-envy-free, every subset of $\mathcal{C}_3^s \setminus \{a_i\}$ contains at least one winner and one loser. Moreover, by Lemma 3.32 we have:

$$\forall a_j \in \mathcal{C}_3^s, j \neq i, \qquad V_j(f_i) < 2\epsilon_j. \tag{21}$$

Note that a_i previously belonged to $\mathcal{C}_3^f$. By definition of $\mathcal{C}_3^f$

$$\forall a_j \in \mathcal{C}_3^s, j \neq i, \qquad V_i(f_j) < 1/2. \tag{22}$$

Inequalities (21) and (22) together imply that a_i is both a winner and a loser for every subset of $\mathcal{C}_3^s$ that contains a_i. This means that every subset of $\mathcal{C}_3^s$ contains at least one winner and one loser, which is a contradiction.

□

Proof of Lemma 3.34:

If $a_j \prec_{pr} a_i$, then g_i is not feasible for a_j, since the agent with the lowest priority is satisfied in each round of the second phase. Thus, $V_j(g_i) < \epsilon_j$. For the case where $a_i \prec_{pr} a_j$, let b_k be an arbitrary item of g_i. According to the fact that g_i is minimal, $g_i \setminus \{b_k\}$ is not feasible for any agent. Hence, $V_j(g_i \setminus \{b_k\}) < \epsilon_j$. On the other hand, by Observations 3.3 and 3.4, $V_j(\{b_k\}) < \epsilon_j$. Therefore, $V_j(g_i) < 2\epsilon_j$. □

Proof of Lemma 3.35: Let $\mathbb{R}_z$ be the round, in which a_i is satisfied. At that point, if $a_j \in \mathcal{C}_3^f$ then $V_j(g_i) < 1/2$ trivially holds. Since in round $\mathbb{R}_z$, $a_j \prec_{pr} a_i$ holds, g_i was not feasible for a_j in the first place. Recall that in each round, the agent with lowest order in $\Phi(S)$ is selected.

Furthermore, if in round $\mathbb{R}_z$, a_j was in $\mathcal{C}_3^s \cup \mathcal{C}_3^b$, according to Observations 3.3 and 3.4, $|S| \geq 2$, since no item alone can satisfy a_i. If $|S| = 2$, then by Observation 3.26, $V_j(g_i) < 1/2$. For the case of $|S| > 2$, let b_k be the item in S with the minimum value to a_j. According to Corollary 3.27, $V_j(\{b_k\}) < 1/4$. Also, since S is minimal, $S \setminus \{b_k\}$ is not feasible for any agent and hence, $V_j(S \setminus \{b_k\}) < \epsilon_j \leq 1/4$. Thus, $V_j(S) < 1/2$. □

E Omitted Proofs of Section 3.5

Before proceeding to the proof of Lemma 3.36, we show Lemmas (E.1, E.2 and E.3).

Lemma E.1 *Let a_i be an agent in $\mathcal{S}_3$ and let $\mathbb{R}_z$ be the round of the second phase in which a_i is satisfied. Then, for any other agent a_j that is in $\mathcal{C}_3^f$ in $\mathbb{R}_z$, $V_j(g_i) < 1/2$ holds.*

Proof. In $\mathbb{R}_z$, a_i either belongs to $\mathcal{C}_3^s$ or $\mathcal{C}_3^b$. Thus, $a_j \prec_{pr} a_i$, and thus g_i is not feasible for a_j in that round. Therefore, $V_j(g_i) < 1/2$. □

Lemma E.2 *Let $a_i \in \mathcal{S}_3$ be a satisfied agent and let $\mathbb{R}_z$ be the round in which a_i is satisfied. Then, for every other agent a_j that belongs to $\mathcal{C}_3^s \cup \mathcal{C}_3^b$ in that round, either $V_j(g_i) < \epsilon_j$ or $V_j(f_i) \leq 3/4 - \epsilon_j$.*

Proof. If g_i is not feasible for a_j, then the condition trivially holds. Moreover, by the definition, the statement is correct for the agents of $\mathcal{C}_3^b$. Therefore, it only suffices to consider the case that $a_j \in \mathcal{C}_3^s$ and g_i is feasible for a_j. Due to the priority rules for satisfying the agents in the second phase, $a_i \prec_{pr} a_j$ and hence, a_i cannot be in $\mathcal{C}_3^b$. Thus, $a_i \in \mathcal{C}_3^s$. According to Observation 3.1 and the fact that $\prec_{pr}$ is equivalent to $\prec_o$ for the agents in $\mathcal{C}_3^s$, we have $V_j(f_i) \leq 3/4 - \epsilon_j$. □

Lemma E.3 *During the second phase, for any agent a_i in $\mathcal{C}_3$, we have:*

$$\sum_{a_j \in \mathcal{S}_3} V_i(f_j \cup g_j) < |\mathcal{S}_3| + 1/4.$$

Proof. To show Lemma E.3, we show that for all the agents $a_j \in \mathcal{S}_3$ except at most one agent, $V_i(f_j \cup g_j) < 1$ holds. To show this, let $\mathbb{R}_z$ be an arbitrary round of the second phase, in which an agent $a_j \in \mathcal{C}_3$ is satisfied. First, note that in $\mathbb{R}_z$, a_j belongs to $\mathcal{C}_3^s \cup \mathcal{C}_3^b$. Also, in round $\mathbb{R}_z$, a_i belongs to one of $\mathcal{C}_3^s, \mathcal{C}_3^b$, or $\mathcal{C}_3^f$.

If $a_i \in \mathcal{C}_3^f$, then by Lemma E.1, $V_i(g_j) < 1/2$ holds. On the other hand, by definition, $V_i(f_j) < 1/2$ and hence, $V_i(f_j \cup g_j) < 1$.

Now, consider the case, where $a_i \in \mathcal{C}_3^b \cup \mathcal{C}_3^s$. Note that by Lemma E.2, either $V_i(f_j) \leq 3/4 - \epsilon_i$ or $V_i(g_j) < \epsilon_i$. If $V_i(g_j) < \epsilon_i$, then by Lemmas 3.28 and 3.32, we know $V_i(f_j) < 3/4$ and hence, $V_i(f_j \cup g_j) < 3/4 + \epsilon_i < 1$.

For the case where $V_i(f_j) \leq 3/4 - \epsilon_i$, let b_l be the item in g_j with the maximum value to a_i. By minimality of g_j, $g_j \backslash \{b_l\}$ is not feasible for any agent, including a_i and thus, $V_i(g_j \backslash \{b_l\}) < \epsilon_i$. Recall that by Corollary 3.27, there is at most one item b_k in $\mathcal{F}$, such that $V_i(b_k) \geq 1/4$. In addition to this, $V_i(b_k) < 1/2$ trivially holds, since b_k is not assigned to any agent during the clustering phase. If $b_l \neq b_k$, $V_i(g_j) < 1/4 + \epsilon_i$ holds and hence,

$$V_i(f_j \cup g_j) < 3/4 - \epsilon_i + 1/4 + \epsilon_i < 1.$$

Moreover, If $b_l = b_k$, $V_i(g_j) < 1/2 + \epsilon_i$ holds and thus, $V_i(f_j \cup g_j) < 3/4 - \epsilon_i + 1/2 + \epsilon_i < 5/4$. But, this can happen at most one round. Therefore, for all the agents $a_j \in \mathcal{S}_3$ except at most one, $V_i(f_j \cup g_j) < 1$. Also, for at most one agent $a_j \in \mathcal{S}_3$, $V_i(f_j \cup g_j) < 5/4$. Thus,

$$\sum_{a_j \in \mathcal{S}_3} V_i(f_j \cup g_j) < |\mathcal{S}_3| + 1/4.$$

□

Proof of Lemma 3.36: Suppose for the sake of contradiction that $\mathcal{C}_3 \neq \emptyset$. Note that, by the definition of $\mathcal{C}_3^b$, if $\mathcal{C}_3^s = \emptyset$ holds, then consequently $\mathcal{C}_3^b = \emptyset$. Therefore, since we have $\mathcal{C}_3 = \mathcal{C}_3^s \cup \mathcal{C}_3^b \cup \mathcal{C}_3^f$, if $\mathcal{C}_3$ is non-empty, at least either of the two sets $\mathcal{C}_3^s$ or $\mathcal{C}_3^f$ is non-empty. In case $\mathcal{C}_3^s$ is non-empty, let a_i be a winner of $\mathcal{C}_3^s$, otherwise let a_i be an arbitrary agent of $\mathcal{C}_3^f$.

According to Lemma 3.35, for every agent $a_j \in \mathcal{S}_1^s \cup \mathcal{S}_2^s$, $V_i(g_j) < 1/2$ holds. Also, by Lemmas 3.16 and 3.25, for every agent $a_j \in \mathcal{S}_1^r \cup \mathcal{S}_2^r$, we have $V_i(g_j) < 1/2$. Therefore,

$$\forall a_j \in \mathcal{S}_1 \cup \mathcal{S}_2 \qquad V_i(g_j) < 1/2.$$

Also, by Lemmas 3.14 and 3.23 we know that $V_i(f_j) < 1/2$ for every $a_j \in \mathcal{S}_1 \cup \mathcal{S}_2$. Thus, for every satisfied agent $a_j \in \mathcal{S}_1 \cup \mathcal{S}_2$, $V_i(f_j \cup g_j) < 1$ holds, and hence

$$\sum_{a_j \in \mathcal{S}_1 \cup \mathcal{S}_2} V_i(f_j \cup g_j) < |\mathcal{S}_1 \cup \mathcal{S}_2|. \tag{23}$$

Moreover, by Lemma E.3, the total value of items assigned to the agents in $\mathcal{S}_3$ to a_i is less than $|\mathcal{S}_3| + 1/4$. More precisely,

$$\sum_{a_j \in \mathcal{S}_3} V_i(f_j \cup g_j) \leq |\mathcal{S}_3| + 1/4. \tag{24}$$

Inequality (23) along with Inequality (24) implies:

$$\begin{aligned} \sum_{a_j \in \mathcal{S}} V_i(f_j \cup g_j) &= \sum_{a_j \in \mathcal{S}_1 \cup \mathcal{S}_2} V_i(f_j \cup g_j) + \sum_{a_j \in \mathcal{S}_3} V_i(f_j \cup g_j) \\ &< |\mathcal{S}_1 \cup \mathcal{S}_2| + |\mathcal{S}_3| + 1/4 \\ &= |\mathcal{S}| + 1/4 \end{aligned} \tag{25}$$

Recall that the total sum of the item values for a_i is equal to n. In addition to this, since every agent belongs to either of the Clusters $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, or $\mathcal{S}$ we have

$$|\mathcal{S}| + |\mathcal{C}_1| + |\mathcal{C}_2| + |\mathcal{C}_3| = n.$$

Furthermore, every item $b_j \in \mathcal{M}$ either belongs to $\mathcal{F}$ or one of the sets $f_{j'}$ and $g_{j'}$ for an agent $a_{j'}$. More precisely,

$$\mathcal{F} = \mathcal{M} \setminus \Big[\bigcup_{a_j \in \mathcal{S} \cup \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3^s} f_j \cup \bigcup_{a_j \in \mathcal{S}} g_j \Big].$$

Therefore

$$\begin{aligned} \sum_{a_j \in \mathcal{C}_1} V_i(f_j) + \sum_{a_j \in \mathcal{C}_2} V_i(f_j) + \sum_{a_j \in \mathcal{C}_3^s} V_i(f_j) + V_i(\mathcal{F}) &= V_i(\mathcal{M}) - \sum_{a_j \in \mathcal{S}} V_i(f_j \cup g_j) \\ &= n - \sum_{a_j \in \mathcal{S}} V_i(f_j \cup g_j) \\ &\geq n - (|\mathcal{S}| + 1/4) \\ &= |\mathcal{C}_1| + |\mathcal{C}_2| + |\mathcal{C}_3| - 1/4 \end{aligned} \tag{26}$$

According to Lemmas 3.14 and 3.17,

$$\sum_{a_j \in \mathcal{C}_1} V_i(f_j) < 1/2|\mathcal{C}_1| \tag{27}$$

and

$$\sum_{a_j \in \mathcal{C}_2} V_i(f_j) < 1/2|\mathcal{C}_2| \tag{28}$$

hold. Inequalities (26), (27), and (28) together prove

$$\begin{aligned} V_i(\mathcal{F}) &\geq |\mathcal{C}_1| + |\mathcal{C}_2| + |\mathcal{C}_3| - 1/4 - \big[\sum_{a_j \in \mathcal{C}_1} V_i(f_j) + \sum_{a_j \in \mathcal{C}_2} V_i(f_j) + \sum_{a_j \in \mathcal{C}_3^s} V_i(f_j) \big] \\ &\geq |\mathcal{C}_1| + |\mathcal{C}_2| + |\mathcal{C}_3| - 1/4 - \big[1/2|\mathcal{C}_1| + 1/2|\mathcal{C}_2| + \sum_{a_j \in \mathcal{C}_3^s} V_i(f_j) \big] \\ &\geq 1/2|\mathcal{C}_1| + 1/2|\mathcal{C}_2| + |\mathcal{C}_3| - 1/4 - \sum_{a_j \in \mathcal{C}_3^s} V_i(f_j). \end{aligned} \tag{29}$$

Now, we consider two cases separately: (i) $a_i \in \mathcal{C}_3^s$ and (ii) $a_i \in \mathcal{C}_3^f$.

In case $a_i \in \mathcal{C}_3^s$, since a_i is a winner of $\mathcal{C}_3^s$, we have

$$\begin{aligned} \sum_{a_j \in \mathcal{C}_3^s} V_i(f_j) &\leq \sum_{a_j \in \mathcal{C}_3^s} V_i(f_i) \\ &= \sum_{a_j \in \mathcal{C}_3^s} 3/4 - \epsilon_i \\ &= (3/4 - \epsilon_i)|\mathcal{C}_3^s|. \end{aligned} \tag{30}$$

This combined with Inequality (29) concludes

$$\begin{aligned} V(\mathcal{F}) &\geq 1/2|\mathcal{C}_1| + 1/2|\mathcal{C}_2| + |\mathcal{C}_3| - 1/4 - \sum_{a_j \in \mathcal{C}_3^s} V_i(f_j) \\ &\geq 1/2|\mathcal{C}_1| + 1/2|\mathcal{C}_2| + |\mathcal{C}_3| - 1/4 - (3/4 - \epsilon_i)|\mathcal{C}_3^s| \\ &\geq 1/2|\mathcal{C}_1| + 1/2|\mathcal{C}_2| + (1/4 + \epsilon)|\mathcal{C}_3| - 1/4. \end{aligned}$$

On the other hand, since $a_i \in \mathcal{C}_3^s$, $|\mathcal{C}_3| \geq 1$ and hence, $V_i(\mathcal{F}) \geq 1/4 + \epsilon_j - 1/4 = \epsilon_j$. This means that $\mathcal{F}$ is feasible for a_i, which contradicts the termination of the algorithm.

In case $a_i \in \mathcal{C}_3^f$, by the definition of $\mathcal{C}_3^f$ we know that $\sum_{a_j \in \mathcal{C}_3^s} V_i(f_j) < 1/2|\mathcal{C}_3^s|$, which by Inequality (29) implies:

$$V_i(\mathcal{F}) > 1/2|\mathcal{C}_3^s| + |\mathcal{C}_3^b| + |\mathcal{C}_3^f| + 1/2|\mathcal{C}_2| + 1/2|\mathcal{C}_1| - 1/4.$$

Since $a_i \in \mathcal{C}_3^f$, we have $|\mathcal{C}_3^f| \geq 1$ and hence, $V_i(\mathcal{F}) > 3/4$. Again, this contradicts the termination of the algorithm since $\mathcal{F}$ is feasible for a_i. □

Proof of Lemma 3.37: By Lemma 3.36, we already know $\mathcal{C}_3 = \emptyset$. Now, let a_i be a winner of the remaining agents in $\mathcal{C}_1$. For convenience, we color the items in either blue or white. Intuitively, blue items may have a high value for a_i whereas white items are always of lower value to a_i. Initially, all items are colored in white. For each $a_j \in \mathcal{N}$, if $|f_j| = 1$, then we color the item in f_j in blue. Moreover, for every $a_j \in \mathcal{S}$, if $|g_j| = 1$ and $V_i(g_j) \geq \epsilon_i$, then we color the item in g_j in blue.

Now, let $\mathcal{P} = \langle P_1, P_2, \ldots, P_n \rangle$ be the optimal n-partitioning of the items in $\mathcal{M}$ for a_i, that is, the value of every partition P_k to a_i is at least 1. Based on the coloring procedure, we have three types of partitions in $\mathcal{P}$:

- B_2: the set of partitions with at least two blue items
- B_1: the set of partitions with exactly one blue item
- B_0: the set of partitions without any blue items

Note that every partition in $\mathcal{P}$ belongs to one of B_0, B_1 or B_2. Hence,

$$|B_0| + |B_1| + |B_2| = n \tag{31}$$

As declared, all the items in the partitions of B_0 are white. The total value of these items to a_i is at least $|B_0| \geq 4\epsilon_i |B_0|$, which is

$$\sum_{P_k \in B_0} \sum_{b_j \in P_k} V_i(b_j) \geq 4\epsilon_i |B_0|. \tag{32}$$

Also, each partition in B_2 has at least two blue items, each of which is singly assigned to another agent. We decompose the partitions of B_1 into two disjoint sets, namely $\hat{B}_1$ and $\tilde{B}_1$. More precisely, let $\hat{B}_1$ be the partitions in B_1, in which the blue item is worth more than $V_i(f_i)$ to a_i and $\tilde{B}_1 = B_1 \setminus \hat{B}_1$. As such, for each partition $P_k \in \tilde{B}_1$, the white items in P_k are worth at least

$$\begin{aligned} 1 - V_i(f_i) &= 1 - (3/4 - \epsilon_i) \\ &= 1/4 + \epsilon_i \\ &\geq 2\epsilon_i \end{aligned}$$

to a_i. Therefore,

$$\sum_{P_k \in \tilde{B}_1} V_i(\mathcal{W}(P_k)) \geq 2|\tilde{B}_1|\epsilon_i \tag{33}$$

where $\mathcal{W}(S)$ stands for the set of white items in a set S of items. On the other hand, since the problem is 3/4-irreducible, by Lemma 3.1, no item alone is of worth 3/4 to a_i and thus for each partition $P_k \in \hat{B}_1$, the white items in P_k have a value of at least $1/4 \geq \epsilon_i$ to a_i. This implies that

$$\sum_{P_k \in \hat{B}_1} V_i(\mathcal{W}(P_k)) \geq |\hat{B}_1|\epsilon_i. \tag{34}$$

By Inequalities (31),(32), (33), and (34) we have

$$\begin{aligned} V_i(\mathcal{W}(\mathcal{M})) &= \sum_{P_j \in B_0} V_i(\mathcal{W}(P_j)) + \sum_{P_j \in B_1} V_i(\mathcal{W}(P_j)) + \sum_{P_j \in B_2} V_i(\mathcal{W}(P_j)) \\ &\geq \sum_{P_j \in B_0} V_i(\mathcal{W}(P_j)) + \sum_{P_j \in B_1} V_i(\mathcal{W}(P_j)) \\ &\geq \sum_{P_j \in B_0} V_i(\mathcal{W}(P_j)) + \sum_{P_j \in \hat{B}_1} V_i(\mathcal{W}(P_j)) + \sum_{P_j \in \tilde{B}_1} V_i(\mathcal{W}(P_j)) \\ &\geq |B_0|4\epsilon_i + |\hat{B}_1|\epsilon_i + |\tilde{B}_1|2\epsilon_i \\ &\geq |B_0|4\epsilon_i + |B_1|2\epsilon_i - |\hat{B}_1|\epsilon_i \\ &\geq |B_0|4\epsilon_i + |B_1|4\epsilon_i + |B_2|4\epsilon_i - |B_1|2\epsilon_i - |B_2|4\epsilon_i - |\hat{B}_1|\epsilon_i \\ &= (2n - 2|B_2| - |B_1| - |\hat{B}_1|)2\epsilon_i + (|\hat{B}_1|)\epsilon_i \end{aligned} \tag{35}$$

Note that the total value of white items that are assigned to the agents during the algorithm is equal to $V_i(\mathcal{W}(\mathcal{M} \setminus \mathcal{F}))$. The rest of the white items are still in $\mathcal{F}$. Thus, we have

$$V_i(\mathcal{W}(\mathcal{M})) = V_i(\mathcal{W}(\mathcal{M} \setminus \mathcal{F})) + V_i(\mathcal{F}) \tag{36}$$

Now, we provide an upper bound on the value of $V_i(\mathcal{W}(\mathcal{M} \setminus \mathcal{F}))$. As a warm up, one can trivially prove an upper bound of $2\epsilon_i(2n - 1 - |B_1| - 2|B_2|)$ on $V_i(\mathcal{W}(\mathcal{M} \setminus \mathcal{F}))$. This follows from the fact that two sets of items are assigned to any agent and hence we have a total of $2n$ disjoint sets. Among these $2n$ sets, at least one of them is empty (since $g_i = \emptyset$) and at least $|B_1| + 2|B_2|$ of the sets contain a single blue item. On the other hand, by Lemmas 3.19, 3.24, 3.32 and 3.34 every set with white items is of worth at most $2\epsilon_i$ to a_i. Therefore, the total value of the white items in $\mathcal{M} \setminus \mathcal{F}$ to a_i is less than $2\epsilon_i(2n - 1 - |B_1| - 2|B_2|)$ and thus

$$V_i(\mathcal{W}(\mathcal{M} \setminus \mathcal{F})) \leq 2\epsilon_i(2n - 1 - |B_1| - 2|B_2|).$$

However, in order to complete the proof, we need a stronger upper bound on $V_i(\mathcal{W}(\mathcal{M} \setminus \mathcal{F}))$. To this end, we provide the following auxiliary lemma.

Lemma E.4 *Let a_j be an agent such that $|f_j| = 1$ and $V_i(f_j) > V_i(f_i)$. Then, $V_i(g_j) < \epsilon_i$.*

Proof. First, note that if a_j is not satisfied yet, then $g_j = \emptyset$ and therefore $V_i(g_j) < \epsilon_i$. Otherwise, we argue that agent a_j is either satisfied in the second phase, or in the refinement phases of $\mathcal{C}_1$ or $\mathcal{C}_2$.

Consider the case that a_j is satisfied in the second phase. If $a_j \in \mathcal{S}_2^s \cup \mathcal{S}_3$, then by Lemma 3.34, $V_i(g_j) < \epsilon$ holds. Also, if $a_j \in \mathcal{S}_1^s$, considering the fact that a_i envies a_j, $a_i \prec_{pr} a_j$. Thus, by Lemma 3.34, we have $V_i(g_j) < \epsilon_i$.

Next, consider the case that a_j is in $\mathcal{S}_1^r \cup \mathcal{S}_2^r$. Note that the matching of the refinement phase of $\mathcal{C}_1$ preserves the property described in Lemma 3.15. Hence, if a_j belongs to $\mathcal{S}_1^r$, then either $a_j \prec_{pr} a_i$ or there is no edge between y_i and $M_1(y_j)$ in G_1, where $M_1(y_j)$ is the vertex matched with y_j in M_1. If $a_j \prec_{pr} a_i$, according to Observation 3.1, $V_i(f_j) \leq 3/4 - \epsilon_i$ holds. On the other hand, by the definition, if no edge exists between y_i and $M_1(y_j)$ in G_1, $V_i(g_j) < \epsilon_i$. In addition to this, if a_j belongs to $\mathcal{S}_2^r$, according to Lemma 3.24, $V_i(g_j) < \epsilon_i$ holds. Therefore, Lemma E.4 holds for the agents in $\mathcal{S}_1^r \cup \mathcal{S}_2^r$.

□

Note that since matching M of $G_{1/2}$ for building Cluster $\mathcal{C}_1$ is MCMWM according to condition (iii) of Lemma 3.13, there exists no agent a_k, such that $|g_k| = 1$ and $V_i(g_k) > 3/4 - \epsilon_i$. Otherwise, by assigning the item in g_j to a_i instead of the item in f_i, we can increase the total weight of the matching, that contradicts the maximality of M.

According to Lemma E.4, for all the agents a_j with the property that f_j is a blue item that belongs to a partition in $\hat{B}_1$, $V_i(g_j) < \epsilon_i$ holds. The number of such agents is at least $|\hat{B}_1|$. Therefore, the total value of $V_i(\mathcal{W}(\mathcal{M} \setminus \mathcal{F}))$ is less than $2\epsilon_i \cdot (2n - 1 - |B_1| - 2|B_2| - |\hat{B}_1|) + \epsilon_i \cdot |\hat{B}_1|$. Combining the bounds obtained in Observation 35 and Lemma E.4 by Inequality (36), we have:

$$V_i(\mathcal{F}) \geq 2\epsilon_i \cdot (2n - 2|B_2| - |B_1| - |\hat{B}_1|) + \epsilon_i \cdot (|\hat{B}_1|) - 2\epsilon_i \cdot (2n - 1 - |B_1| - 2|B_2| - |\hat{B}_1|) - \epsilon_i \cdot |\hat{B}_1|$$

That is:

$$V_i(\mathcal{F}) \geq 2\epsilon_i$$

This contradicts the fact that the set $\mathcal{F}$ is not feasible for a_i.

□

Proof of Lemma 3.38: Lemmas 3.36 and 3.37 state that at the end of the algorithm, $\mathcal{C}_1 = \mathcal{C}_3 = \emptyset$. Now, let a_i be a winner of $\mathcal{C}_2$. We consider two cases separately: $\epsilon_i \geq 1/8$ and $\epsilon_i < 1/8$.

If $\epsilon_i \geq 1/8$, the proof follows from a similar argument we used to prove Lemma 3.37.

Lemma E.5 *If $\epsilon_i \geq 1/8$, then the following inequality holds:*

$$\sum_{a_j \in \mathcal{S}} V_i(f_j \cup g_j) \leq |\mathcal{S}| + 1/8.$$

Proof. We know $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3$. For every agent a_j in $\mathcal{S}_3$, by Lemmas 3.28 and 3.32, we know $V_i(f_j) < 3/4$. Also, according to Lemma 3.34, $V_i(g_j) < \epsilon_i \leq 1/4$. Therefore,

$$\sum_{a_j \in \mathcal{S}_3} V_i(f_j \cup g_j) \leq \sum_{a_j \in \mathcal{S}_3} (3/4 + 1/4) = |\mathcal{S}_3|. \tag{37}$$

Now, consider an agent $a_j \in \mathcal{S}_1$. Note that by Lemma 3.14, $V_i(f_j) < 1/2$. Also, remark that either $a_j \in \mathcal{S}_1^r$ or $a_j \in \mathcal{S}_1^s$. If $a_j \in \mathcal{S}_1^r$ then according to Lemma 3.16, $V_i(g_j) < 1/2$ holds and hence $V_i(f_j \cup g_j) < 1$. Also, If $a_j \in \mathcal{S}_1^s$, then according to Lemma 3.34, $V_i(g_j) < 2\epsilon_i < 1/2$. Thus, in both cases, $V_i(f_j \cup g_j) < 1$ and hence:

$$\sum_{a_j \in \mathcal{S}_1} V_i(f_j \cup g_j) \leq \sum_{a_j \in \mathcal{S}_1} 1 = |\mathcal{S}_1|. \tag{38}$$

Finally consider a satisfied agent $a_j \in \mathcal{S}_2$. Again, remark that either $a_j \in \mathcal{S}_2^r$ or $a_j \in \mathcal{S}_2^s$ holds.

Consider the case that $a_j \in \mathcal{S}_2^s$. If $a_j \prec_{pr} a_i$, then by Observation 3.1, $V_i(f_j) \leq 3/4 - \epsilon_i$ and by Lemma 3.34, $V_i(g_j) < 2\epsilon_i \leq 1/4 + \epsilon_i$ which means $V_i(f_j \cup g_j) < 1$. Moreover, if $a_i \prec_{pr} a_j$, according to Lemmas 3.28 and 3.34, $V_i(f_j \cup g_j) < 3/4 + \epsilon_i \leq 1$. Thus, we have:

$$\sum_{a_j \in \mathcal{S}_2^s} V_i(f_j \cup g_j) \leq \sum_{a_j \in \mathcal{S}_2^s} 1 = |\mathcal{S}_1| \tag{39}$$

It only remains to investigate the case where $a_j \in \mathcal{S}_2^r$. Note that since a_i is not satisfied in the refinement phase of $\mathcal{C}_2$, if $a_i \prec_{pr} a_j$, then $V_i(g_j) < \epsilon_i \leq 1/4$. Otherwise, we could assign the item in g_j to a_i in the refinement phase of $\mathcal{C}_2$. Also, by Lemma 3.28, $V_i(f_j) < 3/4$ holds which yields $V_i(f_j \cup g_j) < 1$.

Finally, if $a_j \prec_{pr} a_i$, by Observation 3.1 $V_i(f_j) \leq 3/4 - \epsilon_i$ holds. Corollary 3.21 states that there is at most one item b_k with $\mathcal{M}_k \in \mathcal{X}' \setminus \mathcal{X}'_{1/2}$ and $V_i(b_k) \geq 3/8$. Also, note that since b_k belongs to $\mathcal{X}' \setminus \mathcal{X}'_{1/2}$, $V_i(\{b_k\}) < 1/2$ holds. For agent a_j, let b_l be the item that is assigned to a_j in the refinement of $\mathcal{C}_2$, i.e., $g_j = \{b_l\}$. We have

$$V_i(f_j \cup g_j) \leq 3/4 - \epsilon_i + V_i(\{b_l\}).$$

If $b_l \neq b_k$, $V_i(f_j \cup g_j) \leq 3/4 - \epsilon_i + 3/8$ holds which by the fact that $\epsilon_i \geq 1/8$, implies $V_i(f_j \cup g_j) \leq 3/4 - 1/8 + 3/8 \leq 1$. In addition to this, If $b_l = b_k$, $V_i(f_j \cup g_j) \leq 3/4 - 1/8 + 4/8 \leq 1 + 1/8$. But this can happen for at most one agent. Thus, for every agent a_j in $\mathcal{S}_2^r$, $V_i(f_j \cup g_j) \leq 1$ holds and for at most one agent $a_j \in \mathcal{S}_2^r$, $V_i(f_j \cup g_j) \leq 1 + 1/8$. Thus, we have

$$\sum_{a_j \in \mathcal{S}_2^r} V_i(f_j \cup g_j) \leq |\mathcal{S}_2^r| + 1/8. \tag{40}$$

Inequality (40) together with Inequality (39) yields

$$\sum_{a_j \in \mathcal{S}_2} V_i(f_j \cup g_j) \leq |\mathcal{S}_2| + 1/8. \tag{41}$$

Furthermore, by Inequalities (37), (38) and (41) we have

$$\begin{aligned} \sum_{a_j \in \mathcal{S}} V_i(f_j \cup g_j) &= \sum_{a_j \in \mathcal{S}_1} V_i(f_j \cup g_j) + \sum_{a_j \in \mathcal{S}_2} V_i(f_j \cup g_j) + \sum_{a_j \in \mathcal{S}_3} V_i(f_j \cup g_j) \\ &\leq |\mathcal{S}_1| + |\mathcal{S}_2| + 1/8 + |\mathcal{S}_3| \\ &\leq |\mathcal{S}| + 1/8. \end{aligned} \tag{42}$$

□

By Lemma E.5, value of agent a_i for the items assigned to the satisfied agents is less than $|\mathcal{S}| + 1/8$. Recall that $\mathcal{C}_2 = \mathcal{C}_3 = \emptyset$ and hence $|\mathcal{S}| = n - |\mathcal{C}_2|$. Therefore,

$$\sum_{a_j \in \mathcal{S}} V_i(f_j \cup g_j) \leq n - |\mathcal{C}_2| + 1/8. \tag{43}$$

Since a_i is a winner of $\mathcal{C}_2$, for all $a_j \in \mathcal{C}_2$, we have $V_i(f_j) \leq V_i(f_i)$. On the other hand, since the total value of all items for a_i is equal to n we have

$$\begin{aligned} V_i(\mathcal{F}) &= V_i(\mathcal{M}) - \sum_{a_j \in \mathcal{C}_2} V_i(f_j) - \sum_{a_j \in \mathcal{S}} V_i(f_j \cup g_j) \\ &= n - \sum_{a_j \in \mathcal{C}_2} V_i(f_j) - \sum_{a_j \in \mathcal{S}} V_i(f_j \cup g_j) \\ &\geq n - \sum_{a_j \in \mathcal{C}_2} V_i(f_j) - \big[n - |\mathcal{C}_2| + 1/8\big] \\ &= |\mathcal{C}_2| - 1/8 - \sum_{a_j \in \mathcal{C}_2} V_i(f_j). \end{aligned} \tag{44}$$

Also, $V_i(f_i) = 3/4 - \epsilon_i$ holds and $V_i(f_j) \leq V_i(f_i)$ for any $a_j \in \mathcal{C}_2$ follows from the fact that a_i is a winner of $\mathcal{C}_2$. Therefore by Inequality (44) we have

$$\begin{aligned} V_i(\mathcal{F}) &\geq |\mathcal{C}_2| - 1/8 - \sum_{a_j \in \mathcal{C}_2} V_i(f_j) \\ &\geq |\mathcal{C}_2| - 1/8 - \sum_{a_j \in \mathcal{C}_2} V_i(f_i) \\ &= |\mathcal{C}_2| - 1/8 - |\mathcal{C}_2| V_i(f_i) \\ &= |\mathcal{C}_2| - 1/8 - |\mathcal{C}_2|(3/4 - \epsilon_i) \\ &= |\mathcal{C}_2|(1/4 + \epsilon_i) - 1/8. \end{aligned}$$

Recall that by the assumption $\epsilon_i \geq 1/8$ holds. Moreover, $\epsilon_i \leq 1/4$, and thus

$$\begin{aligned} V_i(\mathcal{F}) &\geq |\mathcal{C}_2|(1/4 + \epsilon_i) - 1/8 \\ &\geq |\mathcal{C}_2| 2\epsilon_i - 1/8 \\ &\geq |\mathcal{C}_2| 2\epsilon_i - \epsilon_i \end{aligned}$$

and since $|\mathcal{C}_2| \geq 1$,

$$\begin{aligned} V_i(\mathcal{F}) &\geq |\mathcal{C}_2| 2\epsilon_i - \epsilon_i \\ &\geq 2\epsilon_i - \epsilon_i \\ &\geq \epsilon_i \end{aligned}$$

and thus $\mathcal{F}$ is feasible for a_i. This contradicts the termination of the algorithm.

Next, we investigate the case where $\epsilon < 1/8$. Our proof for this case is similar to the one for $\mathcal{C}_1$. Let $\mathcal{S}_1^r$ be the agents in $\mathcal{S}_1$ that are satisfied in the refinement phase and let

$$\mathcal{M}_1^r = \bigcup_{a_j \in \mathcal{S}_1^r} f_j \cup g_j.$$

Lemma 3.17 states that the maxmin value of the agents in $\mathcal{C}_2 \cup \mathcal{C}_3$ for the items in $\mathcal{M}' = \mathcal{M} \setminus \mathcal{M}_1^r$ is at least 1. More precisely for every $a_j \in \mathcal{C}_2$:

$$\mathsf{MMS}_j^{n-|\mathcal{S}_1^r|}(\mathcal{M} \setminus \mathcal{M}_1^r) \geq 1 \tag{45}$$

We color the items of $\mathcal{M}'$ in one of four colors blue, red, green, or white. Initially, all the items are colored in white. For each agent $a_j \in \mathcal{N} \setminus \mathcal{S}_1^r$, if $|f_j| = 1$, then we color the item in f_j in blue. Also, if $|f_j| = 2$ (which means f_j is corresponding to a merged vertex), color both the elements of f_j in red. In addition to this, if $|g_j| = 1$ then color the item in g_j in green. For any set $S \subseteq \mathcal{M}$, we denote the subset of blue, red, green, and white items in S by $\mathcal{B}(S)$,$\mathcal{R}(S)$, $\mathcal{G}(S)$, and $\mathcal{W}(S)$, respectively. Recall that by Lemma 3.20, every pair of items in red or green are worth less that $3/4$ in total to a_i. In other words,

$$V_i(\{b_j, b_k\}) \leq 3/4.$$

for any two different items $b_j, b_k \in \mathcal{B}(\mathcal{M}) \cup \mathcal{G}(\mathcal{M})$. Also, according to Lemmas 3.34 and 3.32, every set including white items is worth less than $2\epsilon_i < 1/4$ to a_i.

Now, let $n' = n - |\mathcal{S}_1^r|$. Let $\mathcal{P} = \langle P_1, P_2, \ldots, P_{n'} \rangle$ be the optimal $n'-$partitioning of $\mathcal{M}'$ for a_i. Recall that by Inequality (45) the value of every partition in $\mathcal{P}$ is at least 1 for a_i. Based on the number of blue and red items in every partition, we define three sets of partitions:

- B_{00} : Partitions with no red or blue items.
- B_{10} : Partitions with blue items, but without any red items.
- B_{01} : Partitions that contain at least one red item.

Next we prove Lemmas E.6 and E.7 to be used later in the proof.

Lemma E.6 *Let $|\mathcal{G}(B_{00})|$ be the number of green items in the partitions of B_{00}. Then,*

$$V_i(\mathcal{W}(B_{00})) \geq (3|B_{00}| - |\mathcal{G}(B_{00})|) \cdot 1/4.$$

Proof. Let B_{00}^j be the set of partitions in B_{00} that contain exactly j green items. We have:

$$|\mathcal{G}(B_{00})| = \sum_{1 \leq j < \infty} j|B_{00}^j| \geq |B_{00}^1| + 2|B_{00}^2| + \sum_{3 \leq j < \infty} 3|B_{00}^j| \tag{46}$$

Also, we have:

$$3|B_{00}| = \sum_{0 \leq j < \infty} 3|B_{00}^j| = 3|B_{00}^0| + 3|B_{00}^1| + 3|B_{00}^2| + \sum_{3 \leq j < \infty} 3|B_{00}^j| \tag{47}$$

Finally, we argue that the value of white items in B_{00} is at least $|B_{00}^0| + |B_{00}^1| \cdot 1/2 + |B_{00}^3| \cdot 1/4$. This follows from the fact that every green item in $P_k \in B_{00}^1$ has a value less than $1/2$ and by Lemma 3.20, every pair of green items in $P_k \in B_{00}^2$ are worth less than $3/4$ to a_j. According to the fact that the value of every partition P_k is at least 1, we have:

$$V_i(\mathcal{W}(B_{00})) \geq |B_{00}^0| + |B_{00}^1| \cdot 1/2 + |B_{00}^3| \cdot 1/4 = \big(4|B_{00}^0| + 2|B_{00}^1| + |B_{00}^2|\big) \cdot 1/4 \tag{48}$$

According to Equations (46) and (47), we have:

$$3|B_{00}| - |\mathcal{G}(B_{00})| \leq 3|B_{00}^0| + 2|B_{00}^1| + |B_{00}^2| \leq 4|B_{00}^0| + 2|B_{00}^1| + |B_{00}^2| \tag{49}$$

Next we combine Equations (48) and (49) to obtain:

$$V_i(\mathcal{W}(B_{00})) \geq \big(3|B_{00}| - |\mathcal{G}(B_{00})|\big) \cdot 1/4 \tag{50}$$

□

Lemma E.7 $V_i(\mathcal{W}(B_{10})) \geq (2|B_{10}| - |\mathcal{B}(B_{10})| - |\mathcal{G}(B_{10})|) \cdot 1/4$

Proof. First, note that every partition in B_{10} contains at least one blue item. Let B_{10}^w be the partitions in B_{10} that contains exactly one blue item and no green item. The other items in each partition of B_{10}^w, are white. Since the problem is 3/4-irreducible, the value of every blue item to a_i is less than $3/4$ and therefore we have:

$$V_i(\mathcal{W}(B_{10})) \geq |B_{10}^w| \cdot 1/4$$

or

$$4V_i(\mathcal{W}(B_{10})) \geq |B_{10}^w|. \tag{51}$$

Moreover, let $B_{10}^{\bar{w}} = B_{10} \setminus B_{10}^w$. Since every partition in B_{10} contains at least one blue item, every partition in $B_{10}^{\bar{w}}$ contains at least two items with colors blue or green. Thus, we have:

$$|\mathcal{G}(B_{10}^{\bar{w}})| + |\mathcal{B}(B_{10}^{\bar{w}})| \geq 2|B_{10}^{\bar{w}}| \tag{52}$$

Summing up Equations (51) and (52) results in

$$4V_i(\mathcal{W}(B_{10})) + |\mathcal{G}(B_{10}^{\bar{w}})| + |\mathcal{B}(B_{10}^{\bar{w}})| \geq 2|B_{10}^{\bar{w}}| + |B_{10}^{w}|$$

which means:

$$4V_i(\mathcal{W}(B_{10})) \geq 2|B_{10}^{\bar{w}}| - |\mathcal{G}(B_{10}^{\bar{w}})| - |\mathcal{B}(B_{10}^{\bar{w}})| + |B_{10}^{w}|. \tag{53}$$

Morover, we have $|\mathcal{B}(B_{10})| = |\mathcal{B}(B_{10}^{w})| + |\mathcal{B}(B_{10}^{\bar{w}})|$. According to the fact that every partition in B_{10}^{w} contains exactly one blue item, $|\mathcal{B}(B_{10}^{w})| = |B_{10}^{w}|$ and hence, $|\mathcal{B}(B_{10})| = |B_{10}^{w}| + |\mathcal{B}(B_{10}^{\bar{w}})|$. By Equation (53), we have:

$$4V_i(\mathcal{W}(B_{10})) \geq 2|B_{10}^{\bar{w}}| - |\mathcal{G}(B_{10}^{\bar{w}})| - |\mathcal{B}(B_{10})| + |B_{10}^{w}| + |B_{10}^{w}|.$$

Finally by the fact that $2|B_{10}^{w}| + 2|B_{10}^{\bar{w}}| = 2|B_{10}|$, we have:

$$4V_i(\mathcal{W}(B_{10})) \geq 2|B_{10}| - |\mathcal{G}(B_{10}^{\bar{w}})| - |\mathcal{B}(B_{10})|$$

which is:

$$V_i(\mathcal{W}(B_{10})) \geq \big(2|B_{10}| - |\mathcal{B}(B_{10})| - |\mathcal{G}(B_{10}^{\bar{w}})|\big) \cdot 1/4$$

□

For the partitions in B_{01}, we construct a graph $G_{01}\langle V_{01}, E_{01}\rangle$, where every vertex $v_j \in V_{01}$ corresponds to a partition $P_j \in B_{01}$. Consider an agent a_j such that f_j consists of a pair of red items $b_k, b_{k'}$ and let $b_k \in P_l$ and $b_{k'} \in P_{l'}$. We add an edge $(v_l, v_{l'})$ to E_{01}. By the definition of B_{01}, $P_l, P_{l'} \in B_{01}$ holds. Note that b_k and $b_{k'}$ might belong to the same partition, i.e., $P_l = P_{l'}$. In this case, we add a loop to G_{01}. Furthermore, for every item $b_k \in \mathcal{B}(B_{01})$, we add a loop to the vertex v_l, where $b_k \in P_l$.

Next, define R_j as the set of partitions in B_{01}, such that the degree of their corresponding vertices in V_{01} are equal to j. In other words:

$$P_k \in R_j \iff d(v_k) = j$$

Next we prove Lemma E.8.

Lemma E.8 *For R_1, we have:*

$$V_i(\mathcal{W}(R_1)) \geq (2|R_1| - |\mathcal{G}(R_1)|) \cdot 1/4$$

Proof. Consider a partition $P_j \in R_1$. Since $d(v_j) = 1$, P_j contains exactly one red item and no blue item. Thus, other items in P_j are either green or white. We show that

$$|\mathcal{G}(P_j)| + 4.V_i(\mathcal{W}(P_j)) \geq 2. \tag{54}$$

First, argue that if $|\mathcal{G}(P_j)| \geq 2$, then Inequality (54) holds. Also, if $|\mathcal{G}(P_j)| = 0$, then $V_i(\mathcal{W}(P_j)) \geq 1/2$, because the value of the red item in P_j is less than $1/2$ (recall that all the red items correspond to the vertices in $\mathcal{X}' \setminus \mathcal{X}'_{1/2}$). This immediately implies the fact that $4.V_i(\mathcal{W}(P_j)) \geq 2$. Finally, if $|\mathcal{G}(P_j)| = 1$, then by Lemma 3.20, the total value of the green and red items in P_j is less than $3/4$ and hence, $V_i(\mathcal{W}(P_j)) \geq 1/4$ which means $|\mathcal{G}(P_j)| + 4.V_i(\mathcal{W}(P_j)) \geq 2$.

Since Inequality (54) holds for every partition $P_j \in R_1$, we have:

$$\sum_{P_j \in R_1} \big(|\mathcal{G}(P_j)| + 4.V_i(\mathcal{W}(P_j))\big) \geq 2|R_1|$$

Therefore,

$$|\mathcal{G}(R_1)| + 4.V_i(\mathcal{W}(R_1)) \geq 2|R_1|$$

and hence,

$$V_i(\mathcal{W}(R_1)) \geq (2|R_1| - |\mathcal{G}(R_1)|) \cdot 1/4$$

□

Lemma E.9 *For R_2, we have:*

$$V_i(\mathcal{W}(R_2)) \geq (|R_2| - |\mathcal{G}(R_2)|) \cdot 1/4$$

Proof. Let P_j be a partition in R_2. First, we show the following inequality holds:

$$4V_i(\mathcal{W}(P_j)) + |\mathcal{G}(P_j)| \geq 1 \tag{55}$$

By the definition of R_2, degree of v_j is 2. Therefore, P_j contains two red items. Note that the degree of the partitions in B_{01} that contain blue items is at least 3. Thus, P_j contains no blue items. By Lemma 3.20, the total value of the red items in P_j is less than 3/4. The rest of the items in P_j are either green or white. If P_j contains a green item, then Inequality (55) holds. On the other hand, if P_j contains no green items, then $V_i(\mathcal{W}(P_j)) \geq 1/4$ and hence, $4V_i(\mathcal{W}(P_j)) \geq 1$. Therefore, Inequality (55) holds in both cases.

Summing up Inequality (55) for all the partitions in R_2, we have:

$$\sum_{P_j \in R_2} 4V_i(\mathcal{W}(P_j)) + |\mathcal{G}(P_j)| \geq |R_2|$$

which means:

$$4V_i(\mathcal{W}(R_2)) + |\mathcal{G}(R_2)| \geq |R_2|$$

That is:

$$V_i(\mathcal{W}(R_2)) \geq \big(|R_2| - |\mathcal{G}(R_2)|\big) \cdot 1/4$$

□

Putting together Lemmas E.6,E.7,E.8, and E.9 we obtain the following lower bound on the valuation of a_i for all white items:

$$\begin{aligned}
V_i(\mathcal{W}(\mathcal{M}')) &= V_i(\mathcal{W}(B_{00})) + V_i(\mathcal{W}(B_{01})) + V_i(\mathcal{W}(B_{10})) \\
&\geq \Big(3|B_{00}| - |\mathcal{G}(B_{00})|\Big) \cdot 1/4 + \Big(2|B_{10}| - |\mathcal{B}(B_{10})| - |\mathcal{G}(B_{10})|\Big) \cdot 1/4 \\
&\quad + \Big(2|R_1| - |\mathcal{G}(R_1)|\Big) \cdot 1/4 + \Big(|R_2| - |\mathcal{G}(R_2)|\Big) \cdot 1/4 \\
&= \Big((3|B_{00}| + 2|B_{10}| + 2|R_1| + |R_2| - |\mathcal{B}(B_{10})|) - \big(|\mathcal{G}(B_{00})| + |\mathcal{G}(B_{10})| + |\mathcal{G}(R_1)| + |\mathcal{G}(R_2)|\big)\Big) \cdot 1/4 \\
&\geq \Big((3|B_{00}| + 2|B_{10}| + 2|R_1| + |R_2|) - |\mathcal{B}(B_{10})| - |\mathcal{G}(\mathcal{M}')|\Big) \cdot 1/4
\end{aligned} \tag{56}$$

where $|\mathcal{G}(\mathcal{M}')|$ is the total number of green items.

The items in $\mathcal{W}(\mathcal{M}')$ are either allocated to an agent during the second phase, or are still in $\mathcal{F}$. Let $\mathcal{W}_2$ be the white items that are allocated to an agent during the second phase. We have:

$$V_i(\mathcal{W}(\mathcal{M}')) = V_i(\mathcal{W}_2) + V_i(\mathcal{F}) \tag{57}$$

Now, we present an upper bound on the value of $V_i(\mathcal{W}_2)$. First, note that the number of agents in $\mathcal{S} \setminus \mathcal{S}_1^r$ is n'. Each of these n' agents has two sets f_j and g_j, that leaves us $2n'$ sets. Since $g_i = \emptyset$ we know that at least one of these sets is empty. Moreover, of all these $|\mathcal{G}(\mathcal{M}')|$ sets contain a single green item and $|\mathcal{B}(B_{10})| + |E_{01}|$ of the sets contain either a single blue item, or a pair of red items (recall that each edge of G_{01} refers to a blue item or a pair of red items). Therefore, the number of the sets that contain only white items is at most:

$$2n' - 1 - |\mathcal{G}(\mathcal{M}')| - |\mathcal{B}(B_{10})| - |E_{01}|$$

By Lemmas 3.34 and 3.32, the value of every set with white items to a_i is less than $2\epsilon_i < 1/4$ and hence:

$$V_i(\mathcal{W}_2) \leq (2n' - 1 - |\mathcal{G}(\mathcal{M}')| - |\mathcal{B}(B_{10})| - |E_{01}|) \cdot 1/4 \tag{58}$$

Subtracting the lower bound obtained for $V_i(\mathcal{W}(\mathcal{M}'))$ in (56) from the upper bound for $V_i(\mathcal{W}_2)$ in (58) gives us a lower bound on the value of $\mathcal{F}$:

$$\begin{aligned}
V_i(\mathcal{F}) &= V_i(\mathcal{W}(\mathcal{M}')) - V_i(\mathcal{W}_2) \\
&\geq \Big((3|B_{00}| + 2|B_{10}| - |\mathcal{B}(B_{10})| + 2|R_1| + |R_2|) - |\mathcal{G}(\mathcal{M}')|\Big) \cdot 1/4 - V_i(\mathcal{W}_2) \\
&\geq \Big((3|B_{00}| + 2|B_{10}| - |\mathcal{B}(B_{10})| + 2|R_1| + |R_2|) - |\mathcal{G}(\mathcal{M}')|\Big) \cdot 1/4 \\
&\quad - \Big(2n' - 1 - |\mathcal{G}(\mathcal{M}')| - |\mathcal{B}(B_{10})| - |E_{01}|\Big) \cdot 1/4 \\
&= \Big(3|B_{00}| + 2|B_{10}| + 2|R_1| + |R_2| - 2n' + 1 + |E_{01}|\Big) \cdot 1/4 \\
&= \Big(2|B_{00}| + 2|B_{10}| + |B_{00}| + |E_{01}| + 2|R_1| + |R_2| - 2n' + 1\Big) \cdot 1/4
\end{aligned} \tag{59}$$

Next we provide Lemmas E.10, E.12, and E.11 to complete the proof.

Lemma E.10 $|B_{00}| \geq |E_{01}| - |B_{01}|$

Proof. First, note that $|B_{00}| + |B_{10}| + |B_{01}| = n'$. Moreover we have $|\mathcal{B}(B_{10})| + |E_{01}| \leq n'$. To show this Lemma, note that each edge in G_{01} corresponds to the first set of an agent in $\mathcal{S} \setminus \mathcal{S}_1^r$. Also, every blue item in B_{10} corresponds to the first set of an agent in $\mathcal{S} \setminus \mathcal{S}_1^r$. Therefore, the total number of the agents must be more than this number. By the definition of B_{10}, we know that $|\mathcal{B}(B_{10})| \geq |B_{10}|$. Therefore, we have:

$$\begin{aligned}
|B_{00}| + |B_{10}| + |B_{01}| &\geq |\mathcal{B}(B_{10})| + |E_{01}| \\
&\geq |(B_{10})| + |E_{01}|
\end{aligned} \tag{60}$$

This means:

$$|B_{00}| \geq |E_{01}| - |B_{01}|$$

□

Lemma E.11 $|E_{01}| \geq 3/2 \sum_{j\geq 3} |R_j| + |R_2| + |R_1|/2$

Proof. $|E_{01}| = \frac{\sum_{v_j \in V_{01}} d(v_j)}{2} = \frac{\sum_j j|R_j|}{2} \geq 3/2 \sum_{j\geq 3} |R_j| + |R_2| + |R_1|/2.$ □

Lemma E.12 $|B_{00}| \geq \frac{\sum_{j\geq 3} |R_j| - |R_1|}{2}$

Proof. By Lemma E.10, $|B_{00}| \geq |E_{01}| - |B_{01}|$. Furthermore, by Lemma E.11,

$$|E_{01}| \geq 3/2 \sum_{j\geq 3} |R_j| + |R_2| + |R_1|/2.$$

By these two inequalities, we have:

$$|B_{00}| \geq 3/2 \sum_{j\geq 3} |R_j| + |R_2| + |R_1|/2 - |B_{01}| \tag{61}$$

Also, since there is a one-to-one correspondence between B_{01} and V_{01}, $|B_{01}| = |V_{01}|$ holds. By the definition of R_j, we have:

$$|V_{01}| = \sum_j |R_j| \tag{62}$$

By replacing the value obtained for B_{01} from (62) into Inequality (61), we have:

$$\begin{aligned} |B_{00}| &\geq 1/2 \sum_{j\geq 3} |R_j| - |R_1|/2 \\ &= \frac{\sum_{j\geq 3} |R_j| - |R_1|}{2}. \end{aligned} \tag{63}$$

□

By applying Lemmas E.12 and E.11 to Inequality (59) we have:

$$\begin{aligned} V_i(\mathcal{F}) &= \left(2|B_{00}| + 2|B_{10}| + |B_{00}| + |E_{01}| + 2|R_1| + |R_2| - 2n' + 1\right) \cdot 1/4 \\ &\geq \left(2|B_{00}| + 2|B_{10}| + \frac{\sum_{j\geq 3} |R_j| - |R_1|}{2} + 3/2 \sum_{j\geq 3} |R_j| + |R_2| + |R_1|/2 + 2|R_1| + |R_2| - 2n' + 1\right) \cdot 1/4 \\ &= \left(2|B_{00}| + 2|B_{10}| + \sum_{j\geq 3} 2|R_j| + 2|R_2| + 2|R_1| - 2n' + 1\right) \cdot 1/4 \end{aligned}$$

Finally, note that $\sum_{j\geq 3} 2|R_j| + 2|R_2| + 2|R_1| = 2|V_{01}| = 2|B_{01}|$. This, together with the fact that $|B_{00}| + |B_{01}| + |B_{10}| = n'$, yields $V_i(\mathcal{F}) \geq (2n' - 2n' + 1) \cdot 1/4$. This means $V_i(\mathcal{F}) \geq 1/4$ which is a contradiction since $\mathcal{F}$ is feasible for a_i. □

F Omitted Proofs of Section 4

Observation F.1 *$f^x(S) \leq x$ for every given S.*

Observation F.2 *$f^x(S) \leq f(S)$ for every given S.*

Proof Of Lemma 4.4:
First Claim: By definition of submodular functions, for given sets A and B we have:

$$f(A \cup B) \leq f(A) + f(B) - f(A \cap B)$$

We prove that $f^x(.)$ is a submodular function in three different cases:

First Case: Let both $f(A)$ and $f(B)$ be at least x. According to Observation F.1, $f^x(A \cup B)$ and $f^x(A \cap B)$ are bounded by x. Therefore, $f^x(A \cup B) + f^x(A \cap B) \leq 2x$, which yields:

$$f^x(A \cup B) + f^x(A \cap B) \leq f^x(A) + f^x(B)$$

Second Case: In this case one of $f(A)$ and $f(B)$ is at least x. We have $f(A \cup B) \geq x$ and $f(A \cap B)$ is no more than $\max\{f(A), f(B)\}$. As a result $f^x(A \cup B)$ and one of $f^x(A)$ or $f^x(B)$ are equal to x which yields:

$$f^x(A \cup B) + f^x(A \cap B) \leq f^x(A) + f^x(B)$$

Third Case: In this case both $f(A)$ and $f(B)$ are less than x, and $f(A \cap B)$ is less than x too. Since $f^x(A) = f(A)$, $f^x(B) = f(B)$, $f^x(A \cap B) = f(A \cap B)$, according to Observation F.2, $f^x(A \cup B) \leq f(A \cup B)$ holds. Since $f(.)$ is a submodular function, we conclude that:

$$f^x(A \cup B) \leq f^x(A) + f^x(B) - f^x(A \cap B).$$

Second Claim: Since $f(.)$ is an XOS set function, by definition, there exists a finite set of additive functions $\{f_1, f_2, \ldots, f_\alpha\}$ such that

$$f(S) = \max_{i=1}^{\alpha} f_i(S)$$

for any set $S \subseteq \mathsf{ground}(f)$. With that in hand, for a given real number x, we define an XOS set function $g(.)$, and show $g(.)$ is equal to $f^x(.)$.

We define $g(.)$ on the same domain as $f(.)$. Moreover, based on $\{f_1, f_2, \ldots, f_\alpha\}$, we define a finite set of additive functions $\{g_1, g_2, \ldots, g_\beta\}$ that describe g. More precisely, for each set S in domain of $f(.)$ we define a new additive function like g_γ in $g(.)$ as follows: Without loss of generality let f_δ be the function which maximizes $f(S)$. For each $b_i \notin S$ let $g_\gamma(b_i) = 0$. Furthermore, for each $b_i \in S$ if $f(S) \leq x$ let $g_\gamma(b_i) = f_\delta(b_i)$, and otherwise let $g_\gamma(b_i) = \frac{x}{f(S)} f_\delta(b_i)$.

We claim that $g(.)$ is equivalent to $f^x(.)$, which implies $f^x(.)$ is an XOS function. $g(.)$ and $f^x(.)$ are two functions which have equal domains. First, we prove that $g(S) \leq f(S)$ for any given set S. According to construction of $g(.)$, for each additive function in $g(.)$ such g_γ, there is at least one additive function in $f(.)$ such f_δ where $g_\gamma(b_i) \leq f_\delta(b_i)$ for each $b_i \in \mathcal{M}$. Therefore, for any given set S we have:

$$g(S) \leq f(S) \tag{64}$$

Now, according to the construction of $g(.)$, for any given set S where $f(S) \leq x$, we have a function $g_\gamma(S) = f(S)$, and where $f(S) > x$, we have a function $g_\gamma(S) = x$. Therefore, we can conclude that:

$$g(S) \geq f^x(S) \tag{65}$$

For any given set S where $f(S) \leq x$, according to the definition of $f^x(.)$, $f(S) = f^x(S)$, and using Inequalities (64) and (65) we argue that $f^x(S) = g(S)$. Moreover, according to the construction of $g(.)$, $g(S) \leq x$ for any given set S. Therefore, for any given set S where $f(S) > x$, according to the definition of $f^x(.)$ and Inequality (65), $f^x(S) = g(S) = x$. As a result, by considering these two

cases we argue that $f^x(.)$ and $g(.)$ are equivalent, which shows $f^x(.)$ is an XOS function.
Third Claim: In this claim, we use a similar argument to the first claim. By definition of subadditive functions for any given sets A and B, we have:

$$f(A \cup B) \leq f(A) + f(B)$$

We prove that $f^x(.)$ meets the definition of subadditive functions by considering two different cases. In the first case at least one of $f(A)$ and $f(B)$ is at least x, and in the second case both $f(A)$ and $f(B)$ is less than x.

First Case: In this case $f^x(A) + f^x(B)$ is at least x, and since $f^x(S) \leq x$ for any given set S, $f^x(A \cup B) \leq x$. Therefore,

$$f^x(A \cup B) \leq f^x(A) + f^x(B)$$

Second Case: Since $f^x(A \cup B) \leq f(A \cup B)$, $f(A \cup B) \leq f(A) + f(B)$, $f(A) = f^x(A)$, and $f(B) = f^x(B)$, we have:

$$f^x(A \cup B) \leq f^x(A) + f^x(B)$$

□

Proof Of Lemma 4.5: Since $f(.)$ is submodular, according to the definition of submodular functions, for every given sets X and Y in domain of $f(.)$ with $X \subseteq Y$ and every $x \in \mathcal{M} \setminus Y$ we have:

$$f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y) \tag{66}$$

Let $S_i = \{e_1, e_2, \ldots, e_\alpha\}$, $T_0 = \emptyset$, and $T_j = \{e_1, e_2, \ldots, e_j\}$, for every $1 \leq j \leq \alpha$. Since $T_j \subseteq S_i$ for each $0 \leq j \leq \alpha$ and f_i is a submodular function, according to Inequality (66) we have:

$$\sum_{1 \leq j \leq \alpha} f_i(S_i \setminus T_{j-1}) - f_i(S_i \setminus T_j) \geq \sum_{1 \leq j \leq \alpha} f_i(S_i) - f_i(S_i - e_j) \tag{67}$$

Since $f_i(S_i) = \sum_{1 \leq j \leq \alpha} f_i(S_i \setminus T_{j-1}) - f_i(S_i \setminus T_j)$, we can rewrite Inequality (67) for every $1 \leq i \leq k$ as follows:

$$f_i(S_i) \geq \sum_{e \in S_i} f_i(S_i) - f_i(S_i - e) \tag{68}$$

For every $1 \leq i \leq k$ we can rewrite Inequality (68) as follows:

$$\sum_{e \in s_i} f_i(S_i - e) \geq (|S_i| - 1) f_i(S_i) \tag{69}$$

By adding $(|\bigcup S_i| - |S_i|) f_i(S_i)$ to the both sides of Inequality (69), we have:

$$\begin{aligned} (|\bigcup S_i| - |S_i|) f_i(S_i) + \sum_{e \in S_i} f_i(S_i - e) &= \sum_{e \in \bigcup S_i} f_i(S_i \setminus \{e\}) \\ &\geq (|\bigcup S_i| - 1) f_i(S_i) \end{aligned} \tag{70}$$

Since Inequality (70) holds for every $1 \leq i \leq k$, we can sum up both sides of Inequality (70) as follows:

$$\sum_{1 \leq i \leq k} \sum_{e \in \bigcup S_i} f_i(S_i - e) \geq \sum_{1 \leq i \leq k} (|\bigcup S_i| - 1) f_i(S_i) \tag{71}$$

By dividing both sides of Inequality (71) over $1/|\bigcup S_i|$ we obtain:

$$\begin{aligned}\frac{1}{|\bigcup S_i|}(\sum_{e\in\bigcup S_i}\sum_{1\leq i\leq k} f_i(S_i-e)) = \mathbb{E}[\sum_{1\leq i\leq k} f_i(S_i^*)] \\ \geq \sum_{1\leq i\leq k} f_i(S_i)\frac{|\bigcup S_i|-1}{|\bigcup S_i|}.\end{aligned} \tag{72}$$

□

Proof Of Lemma 4.6: Similar to the proof of Lemma 4.5, we use Inequality (66) as a definition of submodular functions. Let $S'_i = S_i \setminus S = \{e_1, e_2, \ldots, e_\alpha\}$, $T_0 = S$, and $T_j = S \cup \{e_1, e_2, \ldots, e_j\}$ for $1 \leq j \leq \alpha$. According to $f(S) < 1/3$, $f(S \cup S'_i) \geq 1$, and Inequality (66) as a definition of sub-modular functions, we have:

$$\begin{aligned}2/3 &< f(S\cup S') - f(S) \\ &= \sum_{1\leq j\leq\alpha} f(T_{j-1}\cup\{e_j\}) - f(T_{j-1}) \\ &\leq \sum_{e\in S'_i} f(S\cup\{e\}) - f(S)\end{aligned} \tag{73}$$

Similar to Inequality (71), we can rewrite Inequality (73) with a summation, since Inequality (73) holds for any $1 \leq i \leq k$.

$$2k/3 < \sum_{1\leq i\leq k}\sum_{e\in S'_i} f(S\cup\{e\}) - f(S) \tag{74}$$

By dividing both sides of Inequality (74) over $1/|\bigcup S_i \setminus S|$ we have:

$$\begin{aligned}\frac{2k/3}{|\bigcup S_i\setminus S|} &< \frac{1}{|\bigcup S_i\setminus S|}(\sum_{1\leq i\leq k}\sum_{e\in S'_i} f(S\cup\{e\}) - f(S)) \\ &= \mathbb{E}[f(S\cup\{e\}) - f(S)]\end{aligned} \tag{75}$$

□

G Omitted Proofs of Section 5

Proof of Lemma 5.2: According to the definition of XOS function, $f(.)$ is an XOS function with a finite set of additive functions $\{g_1, g_2, \ldots, g_\alpha\}$ where $f(S) = \max_{i=1}^{\alpha} g_i(S)$ for any set $S \in \mathsf{ground}(f)$. Let $g_j(.)$ be the additive function which maximizes S. Let $g_j(S_1) = \alpha_1, g_j(S_2) = \alpha_2, \ldots, g_j(S_k) = \alpha_k$, which yields $\beta = \sum \alpha_i$. Since $g_j(S_i) = \alpha_i$, $f(S \setminus S_i) \geq \beta - \alpha_i$. Therefore, we have:

$$\begin{aligned}\sum f(S) - f(S\setminus S_i) &\leq \sum \beta - (\beta - \alpha_i) \\ &= \beta \\ &= f(S)\end{aligned} \tag{76}$$

□

Proof of Lemma 5.3: According to the definition of MMS, we know that a_i can divide items to n sets $\mathcal{P} = \langle P_1, P_2, \ldots, P_n \rangle$ such that $V_i(P_j) \geq 1$ for any P_j. The catch is that a_i can divide each of these n sets to two disjoint sets such that the value of each of these new sets be at least $2/5$ to him. Let $T = \{b_1, b_2, \ldots, b_\gamma\}$ be one of these n sets, and $g_j(.)$ be an additive function which maximizes $V_i(T)$. Let $T_k = \{b_1, b_2, \ldots, b_k\}$ for any $1 \leq k \leq \gamma$. According to Lemma 3.1, since the problem is $1/5$-irreducible, the value of any item is less than $1/5$ to a_i. Therefore, there is a set T_k among T_1 to T_γ where $2/5 \leq g_j(T_k) < 3/5$. Since $g_j(.)$ is one of additive functions of XOS function V_i, we have $V_i(T_k) \geq 2/5$. Moreover, since $g_j(T_k) < 3/5$, $g_j(T \setminus T_k) \geq 2/5$, which yields $V_i(T \setminus T_k) \geq 2/5$. As a conclusion, we can divide each of n sets to two disjoint sets with at least $2/5$ value to a_i.

□