

Fair Allocation of Indivisible Items With Externalities

Mohammad Ghodsi ^{*†}

Hamed Saleh [‡]

Masoud Seddighin ^{*}

Abstract

One of the important yet insufficiently studied subjects in fair allocation is the externality effect among agents. For a resource allocation problem, externalities imply that a bundle allocated to an agent may affect the utilities of other agents.

In this paper, we conduct a study of fair allocation of indivisible goods when the externalities are not negligible. We present a simple and natural model, namely *network externalities*, to capture the externalities. To evaluate fairness in the network externalities model, we generalize the idea behind the notion of maximin-share (MMS) to achieve a new criterion, namely, *extended-maximin-share* (EMMS). Next, we consider two problems concerning our model.

First, we discuss the computational aspects of finding the value of EMMS for every agent. For this, we introduce a generalized form of partitioning problem that includes many famous partitioning problems such as maximin, minimax, and leximin partitioning problems. We show that a $1/2$ -approximation algorithm exists for this partitioning problem.

Next, we investigate on finding approximately optimal EMMS allocations. That is, allocations that guarantee every agent a utility of at least a fraction of his extended-maximin-share. We show that under a natural assumption that the agents are α -self-reliant, an $\alpha/2$ -EMMS allocation always exists. The combination of this with the former result yields a polynomial-time $\alpha/4$ -EMMS allocation algorithm.

1 Introduction

Consider a scenario where there is a collection of m indivisible goods that are to be divided amongst n agents. For a properly chosen notion of fairness, we desire our division to be fair. Motivating examples are dividing the inherited wealth among heirs, dividing assets of a bankrupt company among creditors, divorce settlements, task assignments, etc.

Fair division has been a central problem in Economic Theory. This subject was first introduced in 1948 by Steinhaus [24] in the Polish school of mathematics. The primary model used the metaphor of cake to represent a single divisible resource that must be divided among a set of agents. *Proportionality* is one of the most well-studied notions defined to evaluate the fairness of a cake division protocol. An allocation of a cake to n agents is proportional, if every agent feels that his allocated share is worth at least $1/n$ of the entire cake. Despite many positive results regarding proportionality and other fairness notions (e.g. envy-freeness) in cake-cutting (see among many others, [5, 23, 24, 9, 3, 4]), moving beyond the metaphor of cake the problem becomes more subtle. For example, when the resource is a set of indivisible goods, a proportional allocation is not guaranteed to exist for all instances¹.

^{*}Sharif University of Technology

[†]Institute for Research in Fundamental Sciences (IPM) School of Computer Science

[‡]University of Maryland

¹For example, consider the case that there are two agents and the resource is a single indivisible item.

For allocation of indivisible goods, Budish [8] introduced a new fairness criterion, namely *maximin-share*, that attracted a lot of attention in recent years [1, 22, 18, 12, 25]. This notion is a relaxation of proportionality for the case of indivisible items. Assume that we ask agent i to distribute the items into n bundles, and take the bundle with the minimum value. In such a situation, agent i distributes the items in a way that maximizes the value of the minimum bundle. The maximin-share value of agent i is equal to the value of the minimum bundle in the best possible distribution. Formally, the maximin-share of agent i , denoted by MMS_i , for a set \mathcal{M} of items and n agents is defined as

$$\max_{P=\langle P_1, P_2, \dots, P_n \rangle \in \Pi} \min_j V_i(P_j),$$

where Π is the set of all partitions of \mathcal{M} into n bundles, and $V_i(P_j)$ is the value of bundle P_j to agent i . In a nice paper, Procaccia and Wang [22] show that in some instances, no allocation can guarantee maximin-share to all the agents, but an allocation guaranteeing each agent $2/3$ of his maximin-share always exists. This factor has been recently improved to $3/4$ by Ghodsi et al. [12].

Our goal in this paper is to generalize the maximin-share to the case of the agents with externalities. Roughly speaking, externalities are the influences (costs or benefits) incurred by other parties. For resource allocation problems, externalities imply that the bundle allocated to an agent may affect the utility of the other agents. In this work, we assume that the externalities are positive, which is a common assumption in the literature [15, 7, 20].

There are many reasons to consider externalities in an allocation problem. The goods to be divided might exhibit network effects. For example, the value of an Xbox to an agent increases as more of his friends also own an Xbox, since they can play online. Many merit goods generate positive consumption externalities. In healthcare, individuals who are vaccinated entail positive externalities to other agents around them, since they decrease the risk of contraction. Furthermore, allocating a good to an agent might indirectly affect the utility of his friends since they can borrow it.

The messages of our paper can be condensed as follows: **First**, considering the externalities is important: value of EMMS (the natural generalization we define to adapt MMS to the environment with externalities) and MMS might have a large gap. In fact, we show that even a small amount of influence can result in an unbounded gap between these two notions. Thus, when the externalities are not negligible, methods that guarantee MMS to all the agents might no longer be applicable. **Second**, regarding our model and fairness notion, we can approximately maintain fairness in the environment with externalities. In the next section, we give a more detailed description of our results and the techniques used in the paper.

1.1 Our Results and Techniques

In this paper, we take one step toward understanding the impact of externalities in allocation of indivisible items. We start by proposing a general model to capture the externalities in a fair allocation problem under additive assumptions. Although we present some of our results with regard to this general model, the main focus of the paper is on a more restricted model, namely *network externalities*, where the influences imposed by the agents can be represented by a weighted directed graph. This model is inspired by the well-studied linear-threshold model in the context of network diffusion.

We suggest the *extended-maximin-share* notion (EMMS) to adapt maximin-share to the environment with externalities. Similar to maximin-share, our extension is motivated by the maximin strategy in cut-and-choose games. We discuss two aspects of our notion.

First, we discuss the hardness of computing the value of EMMS_i , where EMMS_i is the extended-maximin-share of agent i . For this, we introduce a generalized form of the partition problem that includes many famous partitioning problems such as maximin, minimax, and leximin partitioning problems. This generalized problem is NP-hard due to a trivial reduction from the partition problem. In Section 4, we propose a $1/2$ -approximation algorithm for computing EMMS_i (Theorem 4.2). In fact, we show that the LPT method, which is a famous greedy algorithm in the context of job scheduling, guarantees $1/2$ -approximation for the general partition problem. We also reveal several structural properties of such partitions.

Second, we consider the approximate α -EMMS allocation problem. That is, allocations that guarantee every agent a utility of at least a fraction α of his extended-maximin-share. We define the property of β -self-reliance and show that when the agents are β -self-reliant, there exists an allocation that guarantees every agent i a utility of at least $\beta/2 \cdot \text{EMMS}_i$ (Theorem 5.1). This is our most technically involved result. The basic idea behind our method is as follows: every agent has an expectation value which estimates the utility that he must gain through the algorithm. Initially, the expectation value of agent i is at least $\text{EMMS}_i/2$. In every step of the algorithm, we choose an agent and allocate him a bundle with value at least as his expectation value. Regarding the bundle allocated to this agent, we decrease the expectation value of the remaining agents. Although the algorithm is simple, the analysis is rather complex and heavily exploits the structural properties of the general partitioning problem. The combination of our existential proof with the $1/2$ -approximation algorithm for computing EMMS yields a polynomial time $\beta/4$ -EMMS allocation algorithm.

Additional Results

Brânzei et al. [7] extend the proportionality to the case of the agents with externalities. Here, we define the *average-share* notion and claim that average-share is a better extension of proportionality in presence of externalities. We justify our claim by considering the implications among extended-maximin-share, average-share, and extended-proportionality.

In interest of space, most of the proofs are deferred to the appendix.

1.2 Related Work

Maximin-share has received a lot of attention over the past few years [22, 12, 1, 13, 18, 11, 25, 2, 6]. The counter-example suggested by Procaccia and Wang [22] refutes the existence of any allocation with the maximin-share guarantee. In addition, Procaccia and Wang propose the first approximation algorithm that guarantees each agent $2/3$ of his maximin-share. Recently, Ghodsi et al. [12] improve the approximation ratio to $3/4$. For the special case of 3 agents, Procaccia and Wang [22] prove that guaranteeing $3/4$ of every agent's maximin-share is always possible. This factor is later improved to $7/8$ by Amanatidis et al. [1] and to $8/9$ by Gourvès and Monnot [13]. Kurokawa et al. [18] show that when the valuations are drawn at random, an allocation with maximin-share guarantee exists with a high probability, and it can be found in polynomial time.

Other works generalize maximin-share for different settings. For example, Farhadi et al. [11] generalize maximin-share for the case of asymmetric agents with different entitlements. They introduce the *weighted-maximin-share* (WMMS) criterion and propose an allocation algorithm with a $1/2$ -WMMS guarantee. Suksompong [25] considers the case that the items must be allocated to groups of agents. Gourvès and Monnot [13] extend maximin-share to the case that the goods collectively received by the agents satisfy a matroidal constraint and propose an allocation with a $1/2$ maximin-share guarantee.

In recent years, considering externalities for different problems has received an increasing attention in computer science [16, 15, 7, 20, 19, 2, 21, 6, 11]. For example, Haghpanah et al. [15] study auction design in the presence of externalities. In a more related work, Brânzei et al. [7] consider externalities in the cake cutting problem. They introduce a model for cake cutting with externalities and generalize classic fairness criteria to the case with externalities. Following this work, other works also consider externalities in fair division. For example, Li et al. [20] study truthful and fair methods for allocating a divisible resource with externalities.

2 Model

Throughout the paper, we assume \mathcal{M} is a set of m indivisible items that must be fairly allocated to a set $\mathcal{N} = [n]$ of agents, where $[n]$ denotes the set $\{1, 2, \dots, n\}$. We introduce our model in Section 2.1 and our fairness criteria in Section 2.2.

2.1 Modeling the Externalities.

We start by proposing a general model to represent the externalities. In the **general externalities** model, we suppose that for every set S of items, $V_{j,i}(S)$ reflects the utility that agent i receives by allocating S to agent j . In this model, there is no restriction on the value of $V_{j,i}(\cdot)$, except that the valuations are additive, i.e., $V_{j,i}(S) = \sum_{b \in S} V_{j,i}(\{b\})$.

We also consider a more restricted model where the externalities are due to the relationships between agents. For example, friends may share their items with a probability which is a function of their relationship. We consider a directed weighted graph G where for every pair of vertices i and j , the weight of edge $(\overrightarrow{j, i})$, denoted by $w_{j,i}$, represents the influence of agent j on agent i . We refer such a graph as *influence graph*. If we allocate item b to agent j , the utility gained by agent i from this allocation would be $V_i(\{b\}) \cdot w_{j,i}$, where V_i is the valuation function of agent i . As an example, consider the influence graph illustrated in Figure 1. For the allocation that allocates S_i to agent i ($1 \leq i \leq 6$), total utility of agent 1 would be $V_1(S_1) \cdot 0.8 + V_1(S_2) \cdot 0.2$. We call such a model the **network externalities** model. Notice that, in this model, the utility of agent i for allocating a set S of items to himself is $V_i(S) \cdot w_{i,i}$. In this paper, we suppose w.l.o.g. that the weights of the edges in the influence graph are normalized, so that for every agent i , $\sum_j w_{j,i} = 1$. Although we prove some of our results for the general externalities model, our main focus is on the network externalities model.

Definition 2.1 *We say agent i is β -self-reliant, if $w_{i,i} \geq \beta$.*

For example in Figure 1, agent 1 is 0.8-self-reliant and agent 5 is 0.55-self-reliant. In real-world situations, we expect β to be a value close to 1. Note that, being β -self-reliant for $\beta \simeq 1$ doesn't mean that we can ignore the externalities (for example, see the instance presented in the proof of Observation 5.1).

Definition 2.2 *For every agent a_i , we define the influence vector of agent i , denoted by $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}]$ as the vector representing the influences of the agents on agent i in the influence graph, in non-decreasing order.*

For example, for the graph in Figure 1, we have $x_4 = [0, 0, 0.1, 0.4, 0.5]$ and $x_5 = [0, 0, 0.2, 0.25, 0.55]$. Note that when agent i is β -self-reliant, we have $x_{i,n} \geq \beta$.

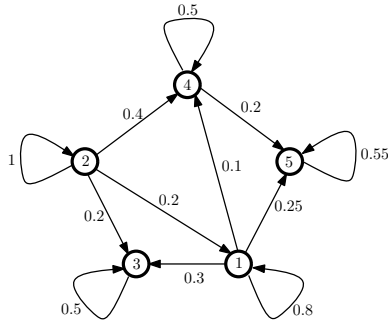


Figure 1: An example of influence graph.

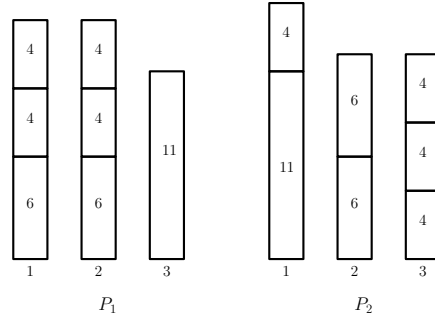


Figure 2: The illustration of two partitions P_1 and P_2 , which are respectively a minimax and a maximin partitions.

2.2 Fairness Criteria

Proportionality and envy-freeness are two of the most common criteria in the literature of cake cutting. For envy-freeness, two extensions are introduced to deal with externalities: *swap-envy-freeness* by Velez [26] and *swap-stability* by Brânzei et al. [7]. In addition, Brânzei et al. [7] defined *extended-proportionality* as follows.

Definition 2.3 (Extended-proportionality [7]) Let \hat{V}_i be the maximum utility that agent i can gain by allocating each item to the right agent, i.e., agent that maximizes the value of that item for agent i . Allocation A is extended-proportional, if the utility of every agent i from A is at least \hat{V}_i/n .

In this paper, we suggest another extension of proportionality, which we call *average-share*.

Definition 2.4 (Average-share) The average value of item b for agent i , denoted by $\bar{V}_i(\{b\})$, is defined as $\sum_j V_{j,i}(\{b\})/n$. The average-share of agent i is $\bar{V}_i(\mathcal{M}) = \sum_{b \in \mathcal{M}} \bar{V}_i(\{b\})$. Furthermore, an allocation is said to be average, if the total utility of every agent from this allocation is at least as his average-share.

It is easy to observe that both of these notions are equivalent to proportionality in the absence of externalities. However, average-share is a stronger notion, i.e., for every agent i , we have $\bar{V}_i(\mathcal{M}) \geq \hat{V}_i/n$. For the network externalities model, we have

$$\hat{V}_i/n = V_i(\mathcal{M}) \cdot (\max_j w_{j,i})/n, \quad , \quad \bar{V}_i(\mathcal{M}) = V_i(\mathcal{M}) \cdot (\sum_j w_{j,i})/n.$$

We claim that average-share is a better extension of proportionality to capture the externalities. Note that extended-proportionality suffers from a drawback, that is its low sensitivity to the externalities. For instance, it is reasonable to assume that the best allocation of agent i is to allocate every item to himself. In such a situation, extended-proportionality completely ignores the externalities. We discuss more on this in Section 3.

It is worth to mention that both the notions described above are too strong to be delivered in the case of indivisible items. For example, when there are no externalities, no allocation can guarantee neither envy-freeness nor proportionality, or even an approximation of them. Thus, no extension of these notions (including extended-proportionality and average-share) can be guaranteed when items are indivisible.

Maximin-share

In this paper, our main focus is on the maximin-share (MMS) criterion. As mentioned, this notion is introduced by Budish [8] as a fairness criteria in division of indivisible items. In Section 1, we gave a formal definition of this notion. The intriguing fact about MMS solution is that it can be motivated by the “cut and choose” game. In this game, an agent divides the items into n bundles and lets other agents choose their bundle first. In the worst-case scenario, the least valued bundle remains, and hence the maximin strategy is to divide the items in a way that the minimum bundle is as attractive as possible. In contrast to proportionality and envy-freeness, guaranteeing a constant fraction of the maximin-share for all the agents is always possible [22, 12].

To extend maximin-share to the case of the agents with externalities, again we consider the worst-case scenario in an “extended cut and choose” game. Suppose that an agent divides the items into n bundles, and other agents somehow distribute these bundles (one bundle to each agent). The maximin strategy of this agent is to divide the items in a way that maximizes his utility in the worst possible scenario (a scenario that minimizes his utility). In fact, we define the *extended-maximin-share* of each agent i as his outcome in the “extended cut and choose” game, regarding maximin strategy.

Formally, let $P = \langle P_1, P_2, \dots, P_n \rangle$ be a partition of \mathcal{M} into n bundles. Furthermore, let $\mathcal{A} : P \rightarrow [n]$ be an allocation function that allocates every set P_i to agent $\mathcal{A}(P_i)$. For brevity, when P is clear from the context, we use \mathcal{A}_i instead of $\mathcal{A}(P_i)$ to refer to the agent whom P_i is allocated to. Since exactly one bundle must be allocated to each agent, \mathcal{A} is a bijection. The utility of agent i for an allocation \mathcal{A} is : $U_i(\mathcal{A}) = \sum_j V_{\mathcal{A}_j, i}(P_j)$. The worst allocation of P regarding agent i , denoted by $\mathcal{W}_i(P)$, is the allocation of P that minimizes the utility of agent i : $\mathcal{W}_i(P) = \arg \min_{\mathcal{A} \in \Omega_P} U_i(\mathcal{A})$, where Ω_P is the set of all $n!$ different allocations of P . Similarly, the best allocation of P is defined as: $\mathcal{B}_i(P) = \arg \max_{\mathcal{A} \in \Omega_P} U_i(\mathcal{A})$. Finally, the extended-maximin-share of agent i , denoted by EMMS_i , is defined as:

$$\text{EMMS}_i = \max_{P \in \Pi} U_i(\mathcal{W}_i(P)),$$

where Π is the set of all partitions of \mathcal{M} into n subsets. We also define the *optimal partition* of \mathcal{M} regarding agent i , denoted by O_i , as the partition that determines the value of EMMS_i , $O_i = \arg \max_{P \in \Pi} U_i(\mathcal{W}_i(P))$. Throughout the paper, when speaking of the network externalities model, we assume that the bundles in $O_i = \langle O_{i,1}, O_{i,2}, \dots, O_{i,n} \rangle$ are sorted by their decreasing values for agent i , i.e., for all j , $V_i(O_{i,j}) \geq V_i(O_{i,j+1})$. In addition, when the agent is clear from the context, for any partition P we use P_j to refer to the j 'th valuable bundle of P , regarding that agent.

Finally, an α -EMMS fair allocation problem with the externalities is defined as follows: is there an allocation such that every agent i receives a utility of at least $\alpha \cdot \text{EMMS}_i$?

3 Model Evaluation

In Section 2.1, we introduced three notions: extended-proportionality, average-share, and extended-maximin-share. For a better understanding of these notions, here we briefly compare them in the **general externalities** model. We already know that average-share is stronger than extended-proportionality. In Lemma 3.1, we prove the same proposition for extended-maximin-share.

Lemma 3.1 *Average-share is a stronger notion than extended-maximin-share.*

By a similar argument as in the proof of Lemma 3.1, we can show that for an arbitrary partition P , $\bar{V}_i(\mathcal{M}) \leq U_i(\mathcal{B}_i(P))$. Therefore, for any partition P we have $\text{EMMS}_i \leq U_i(\mathcal{B}_i(P))$. Lemma 3.1

states that extended-maximin-share is implied by average-share. However, as we show in Lemma 3.2, there is no implication between extended-proportionality and extended-maximin-share.

Lemma 3.2 *Extended-maximin-share does not imply extended-proportionality, nor vice versa.*

The fact that in the case without externalities, proportionality is stronger than maximin-share (MMS), inspires the idea that average-share is a more appropriate extension of proportionality for the case with externalities. In addition, comparing two scenarios in the proof of Lemma 3.2 reveals that the extended-proportionality has a low sensitivity to the externalities.

In the last part of this section, we show that for $n = 2$, a simple *cut and choose* method guarantees EMMS_i to both the agents. Note that there are instances in which neither extended-proportionality nor average-share can be guaranteed even for two agents.

Lemma 3.3 *For two agents, the following simple cut and choose algorithm yields a 1-EMMS allocation:*

- Ask the first agent to partition the items into his optimal partition O_1 .
- Ask the second agent to allocate O_1 (one bundle to each agent).

4 Computing EMMS

In this section, we study the problem of computing EMMS_i and O_i . A closer look at the model reveals that the challenges to calculate EMMS are twofold. One is to find the worst allocation of a given partition, and the other is to find a partition that maximizes the utility of the worst allocation. In Lemma 4.1 and Observation 4.1, we explore the hardness of these problems for the **general externalities** model. We then focus on the network externalities model and give a constant approximation algorithm for computing EMMS.

Lemma 4.1 *Given a partition $P = \langle P_1, P_2, \dots, P_n \rangle$ of the items in \mathcal{M} , the worst allocation of P regarding agent i can be found in polynomial time.*

Observation 4.1 *Since finding the maximin partition of a set of items is NP-hard [27], finding the optimal partition of m items and n agents with externalities is also NP-hard.*

Woeginger [27] also showed that finding the maximin partition of a set of items without externalities admits a PTAS. However, their method does not directly extend to the case with externalities. To the best of our knowledge, finding an approximately optimal partition for an agent in the general externalities model has not been studied before.

In the case of *network externalities*, our model is easier to deal with. Since the utility of each agent is a convex combination of his valuation, finding the worst allocation $\mathcal{W}_i(P)$ is trivial: consider an n -step allocation algorithm whose every step allocates the most valuable remaining bundle to a currently unallocated agent with the least effect on agent i . Hence,

$$U_i(\mathcal{W}_i(P)) = \sum_j x_{i,j} \cdot V_i(P_j). \quad (1)$$

Recall that x_i (the influence vector of agent i) is non-decreasing, and the bundles in P are sorted in non-increasing order of their values for agent i . This property of the network externalities model allows us to approximate the value of EMMS_i with a constant ratio, using a simple greedy

approach. On top of that, it is possible to infer relations between EMMS and some previously defined partitioning schemes.

Apart from the allocation of bundles, partitioning the items is another challenge to overcome. By definition, an optimal partition is a partition that maximizes Equation (1). Finding an optimal partition for a given vector x is in fact, a generalized form of partitioning problems that includes both maximin and minimax partitions. What happens if we partition the items by one of the famous partitioning schemes such as minimax or maximin? A maximin partition is a partition that maximizes the value of the minimum bundle. It is easy to see that a maximin partition is optimal when $x = [1, 0, \dots, 0]$. Likewise, minimax partition is a partition that minimizes the value of the maximum bundle, and it is the optimal partition when $x = [\frac{1}{n-1}, \frac{1}{n-1}, \dots, \frac{1}{n-1}, 0]$. Another example is the leximin partition. A leximin partition first maximizes the minimum bundle, and subject to this constraint, maximizes the second least valued bundle, and so on. Real-world applications of leximin allocations are recently studied by Kurokawa, Procaccia and Shah [17]. For a small enough ϵ , the optimal partition for vector $x = [\frac{1-\epsilon}{1-\epsilon^n}, \frac{\epsilon-\epsilon^2}{1-\epsilon^n}, \frac{\epsilon^2-\epsilon^3}{1-\epsilon^n}, \dots, \frac{\epsilon^{n-1}-\epsilon^n}{1-\epsilon^n}]$ is a leximin partition. For example, in Figure 2, if we choose x to be $[1, 0, 0]$, maximin is the optimal partition, and if we choose x to be $[\frac{1}{2}, \frac{1}{2}, 0]$, minimax is optimal.

Since none of these partitioning schemes are always optimal, approximating either of them is not desirable. However, the well-known greedy algorithm LPT² provides a partition $L_i = \langle L_{i,1}, L_{i,2}, \dots, L_{i,n} \rangle$ for agent i , such that $U_i(\mathcal{W}_i(L_i))$ is a constant approximation of EMMS_i . LPT is a simple greedy algorithm in the context of job scheduling. This algorithm starts with n empty bundles and iteratively puts the most valuable remaining item into the bundle with the minimum total value. It is proved that the partition provided by LPT is a constant approximation for both maximin and minimax partitions [14, 10].

Theorem 4.2 *For the network externalities model, we have*

$$U_i(\mathcal{W}_i(L_i)) \geq \text{EMMS}_i/2. \quad (2)$$

To prove Theorem 4.2, we label some of the items as *huge*. Huge items are those whose values are at least $\bar{V}_i(\mathcal{M})$. Denote the set of huge items for agent i by \mathcal{H}_i .

Claim 4.1 *For an instance with no huge items we have $V_i(L_{i,n}) \geq \bar{V}_i(\mathcal{M})/2$.*

Since $L_{i,n}$ is the least valued bundle of L_i for agent i , $U_i(\mathcal{W}_i(L_i)) \geq V_i(L_{i,n})$. Furthermore, By lemma 3.1, $\bar{V}_i(\mathcal{M}) \geq \text{EMMS}_i$. Hence, when there is no huge item, regarding Claim 4.1, Inequality (2) holds. Thus, to prove Theorem 4.2, it only suffices to consider the instances with huge items. Note that, when there are huge items in \mathcal{M} , $V_i(L_{i,n}) \geq \bar{V}_i(\mathcal{M})/2$ does not necessarily hold. To cope with such a situation, we need to consider some properties for O_i .

Definition 4.3 *We call a partition P nice for agent i , if no item b in some bundle P_j exists, such that $V_i(P_j) > V_i(\{b\}) > V_i(P_n)$ (recall that $V_i(P_n) = \min_j V_i(P_j)$).*

Claim 4.2 *For any partition P , there exists a nice partition P' , such that $U_i(\mathcal{W}_i(P)) \leq U_i(\mathcal{W}_i(P'))$.*

In the rest of this paper, we focus on the optimal partitions which are nice. Furthermore, it can be easily observed that L_i is also nice.

²Longest processing time

In a nice partition P regarding agent i , any bundle P_j containing a huge item $b \in \mathcal{H}_i$ has no other item. Otherwise, $V_i(P_j) > V_i(\{b\}) \geq \bar{V}_i(\mathcal{M})$. Since $\bar{V}_i(\mathcal{M}) > V_i(P_n)$, this is in contradiction with the niceness of P . This fact about nice partitions (including L_i and O_i) allows us to deal with huge items. We are now ready to prove Theorem 4.2.

Proof of Theorem 4.2. We use induction on the number of agents. For $n = 1$, the statement is trivial. For $n > 1$, if $V_i(L_{i,n}) \geq \bar{V}_i(\mathcal{M})/2$ holds, we have $U_i(\mathcal{W}_i(L_i)) \geq \text{EMMS}_i/2$. Thus, when \mathcal{M} contains no huge item, by Claim 4.1, L_i is a $1/2$ -approximation of O_i . Therefore, it only remains to consider the case that $n > 1$ and $V_i(L_{i,n}) < \bar{V}_i(\mathcal{M})/2$. For this case, we know that \mathcal{M} contains at least one huge item.

Let $B_{\mathcal{H}_i}(O_i)$ and $B_{\mathcal{H}_i}(L_i)$ be the set of the bundles containing huge items in O_i and L_i respectively. We know that $B_{\mathcal{H}_i}(O_i) = B_{\mathcal{H}_i}(L_i)$, as the bundles in $B_{\mathcal{H}_i}(O_i)$ and $B_{\mathcal{H}_i}(L_i)$ do not contain anything but huge items, and each huge item is the only item within its bundle (recall that both O_i and L_i are nice). In addition, $B_{\mathcal{H}_i}(L_i)$ are the $|\mathcal{H}_i|$ most valuable bundles in L_i . Otherwise, a very similar argument as in the proof of Claim 4.1 yields $V_i(L_{i,n}) \geq \bar{V}_i(\mathcal{M})/2$.

Let $\mathcal{W}'_i(O_i)$ be the worst possible allocation of O_i with the constraint that allocates $|\mathcal{H}_i|$ huge items to the $|\mathcal{H}_i|$ agents with the least influence on agent i . By definition, $U_i(\mathcal{W}'_i(O_i)) \geq U_i(\mathcal{W}_i(O_i))$. Moreover, in both $\mathcal{W}'_i(O_i)$ and $\mathcal{W}_i(L_i)$, huge items are allocated to the same set of agents, say $\mathcal{N}_{\mathcal{H}_i}$. Now, consider the sub-instance with items $\mathcal{M} \setminus B_{\mathcal{H}_i}(O_i)$ and agents $\mathcal{N} \setminus \mathcal{N}_{\mathcal{H}_i}$. Note that since $V_i(L_{i,n}) < \bar{V}_i(\mathcal{M})/2$, the set $\mathcal{M} \setminus B_{\mathcal{H}_i}(O_i)$ (and hence, $\mathcal{N} \setminus \mathcal{N}_{\mathcal{H}_i}$) is non-empty.

By the induction hypothesis, for this sub-instance, Inequality (2) holds. Now, adding huge items and their corresponding agents back, increases the utility of agent i by the same amount for both of the allocations. Thus, $U_i(\mathcal{W}_i(L_i)) \geq 1/2 \cdot U_i(\mathcal{W}'_i(O_i)) \geq 1/2 \cdot U_i(\mathcal{W}_i(O_i))$. ■

5 α -EMMS Allocation Problem

In this section, we focus on the allocations that guarantee every agent i an approximation of EMMS_i . We start this section by comparing EMMS_i and MMS_i . Note that the value of MMS is determined by the least valued bundle in the maximin partition, while $\text{EMMS}_i = U_i(\mathcal{W}_i(O_i)) \geq U_i(\mathcal{W}_i(Q_i))$, where Q_i is a maximin partition of agent i . In addition, $U_i(\mathcal{W}_i(Q_i))$ is a convex combination of the values of several bundles with a value of at least MMS_i , and hence $\text{EMMS}_i \geq \text{MMS}_i$ always holds. Furthermore, Observation 5.1 states that the gap between EMMS_i and MMS_i could be unbounded even for the instances with 3 agents.

Observation 5.1 *For any $c \geq 1$, there is an instance with 3 agents, where $\text{EMMS}_1 > c \cdot \text{MMS}_1$.*

It is worth to mention that the proof of Observation 5.1 highlights that even for very few externalities, the gap between EMMS and MMS might be large. Thus, the external effects are not negligible even if the impacts of the parties on each other are small.

Our main result is stated in Theorem 5.1. We show that for the network externalities model when all the agents are α -self-reliant, an $\alpha/2$ -EMMS allocation always exists.

Theorem 5.1 *Let \mathbb{C} be an instance where for every agent i , $w_{i,i} \geq \alpha$. Then, \mathbb{C} admits an $\alpha/2$ -EMMS allocation.*

In the rest of this section, we prove Theorem 5.1 by proposing an $\alpha/2$ -EMMS allocation algorithm for the network externalities model with α -self-reliant agents. For brevity, we name our algorithm *Bundle Claiming* algorithm (BC).

5.1 Bundle Claiming Algorithm (BC)

In this section, we present the ideas and a general description of Bundle Claiming algorithm. First, let us review the definition of EMMS_i . With abuse of notations, we suppose that v_i is a vector representing the values of the bundles in the optimal partition of agent i , i.e., $v_{i,j} = V_i(O_{i,j})$. Recall that the bundles in O_i are sorted by their decreasing values for agent i . Hence, for all $j < n$, we have $v_{i,j} \geq v_{i,j+1}$. Furthermore, by definition, we have $\text{EMMS}_i = \sum_j x_{i,j} v_{i,j}$.

Observation 5.2 *For every k , we have $\sum_{j \geq k} x_{i,j} v_{i,j} \leq v_{i,k}$.*

For example, in an instance with $n = 6$, for $k = 4$, Observation 5.2 yields $v_{i,4} \geq v_{i,4}x_{i,4} + v_{i,5}x_{i,5} + v_{i,6}x_{i,6}$. Observation 5.2 is a direct result of the following two facts: first, for all $j > k$, we have $v_{i,k} \geq v_{i,j}$ and second, $\sum_{j > k} x_{i,j} \leq 1$.

Definition 5.2 *For every agent i , we define ℓ_i to be the expectation level of agent i . Agent i with expectation level ℓ_i , has an expectation value of $v_{i,\ell_i}/2$.*

In the beginning of the algorithm, the expectation level of all the agents are set to 1. Our algorithm consists of n steps. In each step, we find a bundle B with the minimum number of items that meets the expectation of at least one agent. Bundle B meets the expectation of agent i , if $V_i(B) \geq v_{i,\ell_i}/2$. We allocate B to one of the agents whose expectation is met (we say this agent is satisfied). Next, we update the expectation levels of the remaining agents. The updating process is a fairly complex process which we precisely describe in Section 5.2. Roughly speaking, we update the expectation levels in a way that the following property holds during the algorithm:

External-satisfaction property: Let \mathcal{S} be the set of currently satisfied agents. For every remaining agent i with expectation level ℓ_i , there is a partition of the agents in \mathcal{S} into ℓ_i subsets, namely $N_{i,1}, N_{i,2}, \dots, N_{i,\ell_i-1}, N_{i,F}$, such that for all $1 \leq j < \ell_i$, the total set of items allocated to the agents in $N_{i,j}$ is worth at least $v_{i,j}/2$ and at most $v_{i,j}$ to agent i , and the total set of items allocated to the agents in $N_{i,F}$ is worth less than $v_{i,\ell_i}/2$ to agent i .

Note that in the updating process, ℓ_i may increase by more than one unit. However, for every remaining agent i , $\ell_i \leq n$ must also hold. As we show in Section 5.2, during our algorithm, $\ell_i \leq n$ always holds for every agent i . We use the lower-bounds in this property to show that if the external-satisfaction property holds for agent i , total amount of externalities incurred by the satisfied agents is at least $\sum_{k < \ell_i} v_{i,k} x_{i,k}/2$. The fact that EMMS_i is calculated with regard to the worst allocation of O_i is the key to prove this inequality.

Consider one step of the algorithm and suppose that a set B of items is allocated to agent i . Since B met the expectation of agent i , $V_i(B) \geq v_{i,\ell_i}/2$. Furthermore, the utility that agent i gained through the externalities of the satisfied agents is at least $\sum_{k < \ell_i} v_{i,k} x_{i,k}/2$. Assuming that agent i is α -self-reliant, his utility is at least

$$\begin{aligned}
& \sum_{k < \ell_i} v_{i,k} x_{i,k}/2 + \alpha v_{i,\ell_i}/2 \\
& \geq \sum_{k < \ell_i} v_{i,k} x_{i,k}/2 + \alpha/2 \sum_{k \geq \ell_i} v_{i,k} x_{i,k} && \text{(Observation 5.2)} \\
& \geq \alpha/2 \sum_k v_{i,k} x_{i,k} && (\alpha \leq 1) \\
& = \alpha/2 \text{EMMS}_i. && (3)
\end{aligned}$$

ALGORITHM 1: Bundle Claiming algorithm

```
1 forall the  $a_j \in \mathcal{N}$  do
2    $\ell_j \leftarrow 1$  ▷ Initializing expectation levels
3 while  $\mathcal{N} \neq \emptyset$  do
4   forall the  $a_j \in \mathcal{N}$  do
5      $\Gamma_j \leftarrow$  Minimum sized subset of  $\mathcal{M}$ , s.t.  $V_j(\Gamma_j) \geq 1/2 \cdot v_{j,\ell_j}$ 
6     Allocate the minimum sized  $\Gamma_i$  to agent  $i$ . ▷ Allocating
7     Remove agent  $i$  from  $\mathcal{N}$ , and  $\Gamma_i$  from  $\mathcal{M}$ .
8   forall the  $a_j \in \mathcal{N}$  do
9     Add agent  $i$  to  $N_{j,F}$ .
10    while  $V_j(N_{j,F}) \geq 1/2 \cdot v_{j,\ell_j}$  do
11      Update  $M_j$ . ▷ Maintaining external-satisfaction (see Section 5.2)
```

Inequality (3) ensures that the items allocated to agent i satisfy him. Furthermore, we use the upper bounds in the external-satisfaction property to prove that the algorithm satisfies all the agents. To show this, it only suffices to prove that in each step of the algorithm there are enough items to meet the expectation of the remaining agents. Consider agent i which has not satisfied yet. The value of the items allocated to the satisfied agents not in $N_{i,F}$ is at most $\sum_{j < \ell_i} v_{i,j}$. Hence, the total value of the remaining items plus the items allocated to the agents in $N_{i,F}$ is at least $\sum_{j \geq \ell_i} v_{i,j}$. On the other hand, the value of the items allocated to the agents in $N_{i,F}$ is less than $v_{i,\ell_i}/2$. Thus, the value of the remaining items is at least $v_{i,\ell_i}/2$ which is enough to meet the expectation of agent i . As said before, our algorithm maintains the property that $\ell_i \leq n$ for every remaining agent i .

The details of the bundle claiming algorithm is demonstrated in Algorithm 1, which is a high level abstraction of the full algorithm and captures the overall sketch of the algorithm. In the next section, we show how to maintain the external-satisfaction property in the algorithm.

We end this section by showing that BC can be implemented in polynomial time. The only part of the algorithm whose implementation in polynomial time is not trivial is when we want to find a minimum-sized set B meeting the expectation of at least one agent.

Observation 5.3 *The minimum-sized set that meets the expectation of at least one remaining agent can be found in polynomial time.*

It is worth to mention that the operations we apply in order to maintain the external-satisfaction property in the next section, are also trivially polynomial time.

Finally, using L_i ³ instead of O_i in BC results in an $\alpha/4$ -EMMS allocation algorithm.

Corollary 5.3 *Let \mathbb{C} be an instance where for every agent i , $w_{i,i} \geq \alpha$. Then, an $\alpha/4$ -EMMS allocation for \mathbb{C} can be found in polynomial time.*

5.2 Maintaining the External-satisfaction Property

Throughout this section, we suppose that \mathcal{S} is the set of satisfied agents. Furthermore, for each agent i in \mathcal{S} , we denote the bundle allocated to him by B_i . We start by giving a detailed explanation

³Partitioning provided by LPT algorithm

of the updating process. As mentioned in the previous section, the external-satisfaction condition must hold during the entire algorithm. To maintain this property in the updating process, for every agent i , we define a mapping M_i that represents the partitioning of \mathcal{S} for agent i (recall the definition of external-satisfaction).

Definition 5.4 *For every agent i , we define $M_i : \mathcal{S} \rightarrow \{O_{i,1}, O_{i,2}, \dots, O_{i,n}\} \cup \{F_i\}$ as a mapping that corresponds each satisfied agent to a bundle in the optimal partition of O_i or to F_i . Furthermore, we define $N_{i,j}$ as the set of agents that are mapped to $O_{i,j}$ in M_i and $N_{i,F}$ as the set of agents mapped to F_i . During the algorithm, we say mapping M_i is valid, if the following conditions hold:*

- (i) $\forall j < \ell_i \quad \sum_{k \in N_{i,j}} V_i(B_k) \geq v_{i,j}/2$
- (ii) $\forall j < \ell_i \quad \sum_{k \in N_{i,j}} V_i(B_k) \leq v_{i,j}$
- (iii) $\sum_{k \in N_{i,F}} V_i(B_k) < v_{i,\ell_i}/2$

During the algorithm, mapping M_i must remain valid for every unsatisfied agent i . In the beginning, $\mathcal{S} = \emptyset$ and for every agent i , $\ell_i = 1$ and hence, M_i is valid. In every step of the algorithm, we satisfy an agent i by a bundle B_i of items. Next, for every unsatisfied agent j , we map agent i to F_j in M_j , i.e., we set $M_j(i) = F_j$. In fact, $N_{j,F}$ corresponds to the satisfied agents that are not mapped to any bundle of O_j in M_j . We use these agents to update ℓ_j . Throughout the algorithm, whenever the total value of the items allocated to the agents in $N_{j,F}$ reaches $v_{j,\ell_j}/2$, M_j becomes invalid and hence, we need to update ℓ_j and M_j to reinstate the validity of M_j . To do so, we pick a subset δ of the agents in $N_{j,F}$ with the minimum size to map them to O_{j,ℓ_j} . Regarding the validity conditions of M_j , total value of the items allocated to the agents in δ must be at least $v_{j,\ell_j}/2$ and at most v_{j,ℓ_j} (we call such subset a *compatible set*). If a compatible set δ exists, we map the agents in δ to O_{j,ℓ_j} in M_j and increase ℓ_j by one. However, there may be some cases that no subset of $N_{j,F}$ is compatible. For such cases, we use the argument in Lemma 5.5.

Lemma 5.5 *Suppose that total value of the items allocated to the agents in $N_{j,F}$ is at least $v_{j,\ell_j}/2$, but $N_{j,F}$ admits no compatible subset. Then, it is possible to modify M_j such that conditions (i) and (ii) remain valid for M_j and $N_{j,F}$ contains at least one compatible subset.*

Note that, after increasing ℓ_j for agent j , condition (iii) may still be violated. In that case, as long as condition (iii) is violated, we continue updating. Each time we update M_j , value of ℓ_j is increased by one. Since at least one agent is mapped to $O_{j,\ell}$ for each $\ell < \ell_j$, ℓ_j never exceeds n . In the appendix, you can find a pseudo-code for the updating process (Algorithm 2)

In the last part of this section, we prove Lemma 5.6 which shows that the value of the externalities imposed to agent i by the satisfied agents is lower-bounded by $\sum_{j < \ell_i} x_{i,j} v_{i,j}/2$. As said before, the fact that EMMS_i is defined with regard to the worst allocation of O_i plays a key role in proving Lemma 5.6.

Lemma 5.6 *Consider one step of the algorithm, and let agent i be an arbitrary remaining agent with $\ell_i > 1$. Then, we have $\sum_{j \in \mathcal{S}} w_{j,i} \cdot V_i(B_j) \geq \sum_{j < \ell_i} x_{i,j} \cdot v_{i,j}/2$.*

References

- [1] Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. Approximation algorithms for computing maximin share allocations. In *International Colloquium on Automata, Languages, and Programming*, pages 39–51. Springer, 2015.
- [2] Nima Anari, Shayan Ehsani, Mohammad Ghodsi, Nima Haghpanah, Nicole Immorlica, Hamid Mahini, and Vahab S Mirrokni. Equilibrium pricing with positive externalities. In *International Workshop on Internet and Network Economics*, pages 424–431. Springer, 2010.
- [3] Haris Aziz and Simon Mackenzie. A discrete and bounded envy-free cake cutting protocol for any number of agents. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 416–427. IEEE, 2016.
- [4] Xiaohui Bei, Ning Chen, Xia Hua, Biaoshuai Tao, and Endong Yang. Optimal proportional cake cutting with connected pieces. In *AAAI*, volume 12, pages 1263–1269, 2012.
- [5] Steven J Brams and Alan D Taylor. An envy-free cake division protocol. *The American Mathematical Monthly*, 102(1):9–18, 1995.
- [6] Simina Brânzei, Tomasz Michalak, Talal Rahwan, Kate Larson, and Nicholas R Jennings. Matchings with externalities and attitudes. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 295–302. International Foundation for Autonomous Agents and Multiagent Systems, 2013.
- [7] Simina Brânzei, Ariel D Procaccia, and Jie Zhang. Externalities in cake cutting. *AAAI*, 2013.
- [8] Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.
- [9] Sina Dehghani, Alireza Farhadi, MohammadTaghi HajiAghayi, and Hadi Yami. Envy-free chore division for an arbitrary number of agents. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2564–2583. SIAM, 2018.
- [10] Bryan L Deuermeier, Donald K Friesen, and Michael A Langston. Scheduling to maximize the minimum processor finish time in a multiprocessor system. *SIAM Journal on Algebraic Discrete Methods*, 3(2):190–196, 1982.
- [11] Alireza Farhadi, MohammadTaghi Hajiaghayi, Mohammad Ghodsi, Sebastien Lahaie, David Pennock, Masoud Seddighin, Saeed Seddighin, and Hadi 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.
- [12] Mohammad Ghodsi, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods: Improvement and generalization. *arXiv preprint arXiv:1704.00222*, 2017.
- [13] Laurent Gourvès and Jérôme Monnot. Approximate maximin share allocations in matroids. In *International Conference on Algorithms and Complexity*, pages 310–321. Springer, 2017.
- [14] Ronald L. Graham. Bounds on multiprocessing timing anomalies. *SIAM journal on Applied Mathematics*, 17(2):416–429, 1969.

- [15] Nima Haghpanah, Nicole Immorlica, Vahab Mirrokni, and Kamesh Munagala. Optimal auctions with positive network externalities. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 11–20. ACM, 2011.
- [16] David Kempe and Mohammad Mahdian. A cascade model for externalities in sponsored search. *Internet and Network Economics*, pages 585–596, 2008.
- [17] David Kurokawa, Ariel D Procaccia, and Nisarg Shah. Leximin allocations in the real world. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 345–362. ACM, 2015.
- [18] David Kurokawa, Ariel D Procaccia, and Junxing Wang. When can the maximin share guarantee be guaranteed? In *AAAI*, volume 16, pages 523–529, 2016.
- [19] Renato Paes Leme, Vasilis Syrgkanis, and Éva Tardos. Sequential auctions and externalities. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 869–886. SIAM, 2012.
- [20] Minming Li, Jialin Zhang, and Qiang Zhang. Truthful cake cutting mechanisms with externalities: do not make them care for others too much! In *IJCAI*, pages 589–595, 2015.
- [21] Vahab S Mirrokni, Sebastien Roch, and Mukund Sundararajan. On fixed-price marketing for goods with positive network externalities. In *WINE*, pages 532–538. Springer, 2012.
- [22] Ariel D Procaccia and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 675–692. ACM, 2014.
- [23] Jack Robertson and William Webb. *Cake-cutting algorithms: Be fair if you can*. AK Peters/CRC Press, 1998.
- [24] Hugo Steinhaus. The problem of fair division. *Econometrica*, 16(1), 1948.
- [25] Warut Suksompong. Approximate maximin shares for groups of agents. *arXiv preprint arXiv:1706.09869*, 2017.
- [26] Rodrigo Velez. Fairness and externalities. *Unpublished Manuscript, University of Rochester*, 2008.
- [27] Gerhard J Woeginger. A polynomial-time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters*, 20(4):149–154, 1997.

A Missing Proofs

Proof of Lemma 3.1. Consider O_i and let $\mathbb{A} = \{A^1, A^2, \dots, A^n\}$, where A^k is an allocation of O_i that allocates $O_{i,j}$ to agent $j' = ((j + k - 1) \bmod n) + 1$. Since in \mathbb{A} each item is allocated to each agent once, $\sum_j U_i(A^j) = \sum_j V_{j,i}(\mathcal{M})$. Thus, the worst allocation in set \mathbb{A} has a utility of at most $\sum_j V_{j,i}(\mathcal{M})/n = \bar{V}_i(\mathcal{M})$ for agent i . As a result, $\text{EMMS}_i = U_i(\mathcal{W}_i(O_i)) \leq \bar{V}_i(\mathcal{M})$. ■

Proof of Lemma 3.2. Consider an instance with one item b and n agents, and the following two scenarios for the externalities:

$$(i) \quad \forall_{i \neq j}, V_{j,i}(\{b\}) = 0$$

$$(ii) \quad \forall_{i \neq j}, V_{j,i}(\{b\}) = V_{i,i}(\{b\})$$

In scenario (i), $\text{EMMS}_1 = 0$, while the extended-proportional share of agent 1 is $V_{1,1}(\{b\})/n$. In the second scenario, $\text{EMMS}_1 = V_{1,1}(\{b\})$, but the extended-proportional share of agent 1 is still $V_{1,1}(\{b\})/n$. ■

Proof of Lemma 3.3. We know $U_2(\mathcal{B}_2(O_1)) \geq \text{EMMS}_2$. Furthermore, since $\mathcal{W}_1(O_1)$ determines the value of EMMS_1 , we have: $U_1(\mathcal{B}_2(O_1)) \geq \text{EMMS}_1$. ■

Proof of Lemma 4.1. Consider a complete bipartite graph $G(X, Y)$ where X represents the bundles of P , and Y represents the agents and there is an edge with weight $V_{j,i}(P_k)$ between every pair $x_k \in X$ and $y_j \in Y$. Finding $\mathcal{W}_i(P)$ is equivalent to finding the maximum cardinality matching with minimum weight in G . Classic network flow algorithms solve this problem in polynomial time [?]. ■

Proof of Claim 4.1. Consider $L_{i,1}$ (the most valuable bundle of L_i for agent i). Trivially, we have $V_i(L_{i,1}) \geq \bar{V}_i(\mathcal{M})$, and since there is no huge item, $L_{i,1}$ contains at least two items. On the other hand, according to LPT, the items within a bundle arrive in non-increasing order. Therefore, the last item added to $L_{i,1}$ has a value of at most $V_i(L_{i,1})/2$ and the total value of $L_{i,1}$ just before the last item arrives must have been at least $V_i(L_{i,1})/2$. Furthermore, whenever an item is added to a bundle, that bundle has the minimum value among all the bundles. Therefore, $V_i(L_{i,n}) \geq V_i(L_{i,1})/2 \geq \bar{V}_i(\mathcal{M})/2$. ■

Proof of Claim 4.2. Since P is not nice, there exists an item b in bundle P_j , such that $V_i(P_j) > V_i(\{b\}) > V_i(P_n)$. We modify P as follows: we remove P_j and P_n from P and add two new bundles $A = \{b\}$ and $B = P_j \cup P_n \setminus \{b\}$ to P . Let l and l' be the indices of the newly added bundles in P (note that the bundles are rearranged by their decreasing values for agent i), such that $j \leq l \leq l' \leq n$ (see Figure 3). We have

$$V_i(P_j) > \max(V_i(A), V_i(B)) \geq \min(V_i(A), V_i(B)) > V_i(P_n).$$

By this modification, $U_i(\mathcal{W}_i(P))$ increases by a value of at least

$$x_{i,l} \cdot (\max(V_i(A), V_i(B)) - V_i(P_j)) + x_{i,l'} \cdot (\min(V_i(A), V_i(B)) - V_i(P_n)),$$

which is non-negative since

$$V_i(P_j) + V_i(P_n) = \max(V_i(A), V_i(B)) + \min(V_i(A), V_i(B)),$$

and $x_{i,l} \leq x_{i,l'}$. Let $\mathcal{L}(P) = \{P_j \mid V_i(P_j) = V_i(P_n)\}$. After each modification, either $V_i(P_n)$ increases, or $|\mathcal{L}(P)|$ decreases. Therefore, sequence $(V_i(P_n), V_i(P_{n-1}), \dots, V_i(P_1))$ increases lexicographically

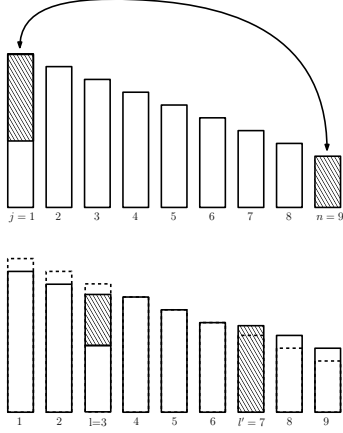


Figure 3: Switching the subsets

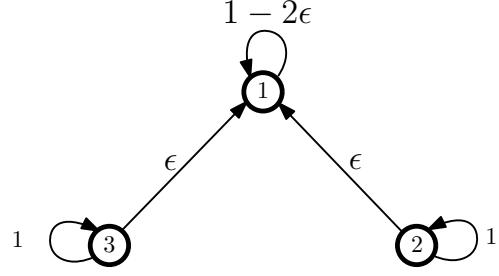


Figure 4: The gap between MMS_i and $EMMS_i$ may be large, even with very small externalities

by each move, and hence we eventually end up with a nice partition P' after a finite number of modifications. ■

Proof of Observation 5.1. Simply consider the influence graph depicted in Figure 4 and three items b_1, b_2 and b_3 such that $V_1(\{b_1\}) = 1$ and $V_1(\{b_2\}) = V_1(\{b_3\}) = c/\epsilon$, where ϵ is a small constant less than $1/2$. For this instance, $EMMS_1 = (1 - 2\epsilon) + 2c$, and $MMS_1 = 1 - 2\epsilon$ which means $MMS_1/EMMS_1 < 1/c$. ■

Proof of Observation 5.3. For every remaining agent i , we find a bundle Γ_i with the minimum size which meets the expectation of agent i as follows: sort the remaining items in their decreasing values for agent i , and add the items to Γ_i one by one until the bundle meets the expectation of agent i . Finally, it only suffices to select the smallest bundle among these bundles. ■

Proof of Lemma 5.5. Let δ be a subset of $N_{j,F}$ with the minimum size that satisfies condition (i). Such a set trivially exists. Since no subset of $N_{j,F}$ is compatible, we have $\sum_{k \in \delta} V_j(B_k) > v_{j,\ell_j}$. By minimality of δ , no proper subset of δ satisfies condition (i). It is easy to observe that this can only happen when δ contains only one agent, say k , with $V_i(B_k) > v_{j,\ell_j}$. We show in Lemma A.1 that $|B_k| = 1$; but for now suppose that b is the only item in B_k .

Since O_j is nice⁴, there is an index $\ell < \ell_j$, such that bundle $O_{j,\ell} = \{b\}$. We modify M_j as follows: we map agent k to $O_{j,\ell}$ and map the former agents of $N_{j,\ell}$ to F_j . Clearly, conditions (i) and (ii) preserve for M_j after this process. Again, if no subset of $N_{j,F}$ is compatible, we repeat this modification. Each time we modify M_j , the number of indices ℓ for which $O_{j,\ell}$ is mapped to an agent j with $O_{j,\ell} = B_j$ increases by one. Therefore, the process terminates after a finite number of modifications. ■

Algorithm 2 illustrates an overview of the update procedure, which completes the BC algorithm.

⁴Recall the niceness from Definition 4.3

ALGORITHM 2: Update M_j

```

1 Resolve = 0
2 while Resolve == 0 do
3    $\delta \leftarrow$  Minimum sized subset of  $N_{j,F}$ , s.t.  $\sum_{k \in \delta} V_j(B_k) \geq 1/2 \cdot v_{j,\ell_j}$ 
4   if  $\sum_{k \in \delta} V_j(B_k) \leq v_{j,\ell_j}$  then
5      $N_{j,\ell_j} = \delta$ 
6      $N_{j,F} = N_{j,f} \setminus \delta$ 
7      $\ell_j \leftarrow \ell_j + 1$ 
8     Resolve+ = 1 ▷ Resolved
9   else
10    Let  $\ell$  be an index s.t.  $O_{j,\ell} = B_k$ , where  $\delta = \{k\}$ .
11    Swap  $\delta$  (which is a subset of  $N_{j,F}$ ) with  $N_{j,\ell}$ . ▷ One step closer to resolve

```

Lemma A.1 *Suppose that the total value of the items allocated to the agents in $N_{j,F}$ is at least $v_{j,\ell_j}/2$, but $N_{j,F}$ admits no compatible subset, and let δ be the minimal subset of $N_{j,F}$ that satisfies condition (i). Then, δ contains only one agent, say agent k and $|B_k| = 1$.*

Proof. As mentioned in Lemma 5.5, it is easy to observe that $|\delta| = 1$. Here, we argue that if agent k is the only agent in δ , then $|B_k| = 1$. As a contradiction, let z_1 be the first step of the algorithm that $\delta = \{k\}$, but $|B_k| > 1$. In addition, let z_2 be the step that B_k was allocated to agent k and let ℓ'_j be the expectation level of agent j in step z_2 . Trivially, we have $z_2 \leq z_1$.

Claim A.1 *Either $v_{j,n} \geq v_{j,\ell'_j}/2$ or we have $|O_{j,\ell}| = 1$ for all $\ell \leq \ell'_j$.*

Proof of Claim A.1. If for some $\ell \leq \ell'_j$, $O_{j,\ell}$ contains more than one item, $O_{j,\ell}$ has a proper subset s such that $V_j(s) \leq v_{j,\ell}/2$. By the same reasoning as Claim 4.2, moving s to bundle $O_{j,n}$ yields a new partition which is at least as good as O_j (See Figure 3). Hence, we can assume w.l.o.g. that $v_{j,n} \geq v_{j,\ell'_j}/2$ holds. ■

Regarding Claim A.1, we consider two cases.

First, assume that $|O_{j,\ell}| = 1$ for all $\ell \leq \ell'_j$. For this case, at least one of the items in $\bigcup_{\ell \leq \ell'_j} O_{j,\ell}$ is not allocated to any agent before step z_2 , and this item singly meets the expectation of agent j . This contradicts the fact that at step z_2 , B_k was the minimal set (Note that we supposed $|B_k| > 1$).

Second, assume that $v_{j,\ell_j} \geq v_{j,\ell'_j}/2$. In step z_2 , the expectation value of agent j equals $v_{j,\ell'}/2$. Furthermore, $V_j(B_k) > v_{j,\ell_j}$ which means $V_j(B_k) > v_{j,\ell'_j}/2$. On the other hand, $V_j(B_k) < v_{j,\ell'_j}$, otherwise a proper subset of B_k would meet the expectation of agent j in step z_2 . Therefore, in step z_2 , $\delta = \{k\}$ is the only compatible set for updating M_j and hence, agent k is mapped to O_{j,ℓ'_j} . This also implies that $z_2 \neq z_1$, since we supposed that no compatible subset exists in step z_1 .

Furthermore, notice that since $|B_k| > 1$, no item could singly meet the expectation of any agent, including agent j in step z_2 . This means that every remaining item in step z_2 has the value less than $v_{j,\ell_j}/2$. On the other hand, in all the modifications before step z_1 , the bundle allocated to the agent in δ consists of only one item (z_1 is the first step that the size of the bundle allocated to the agent in δ is more than 1). This means that after step z_2 , no modification affects the agents that are mapped to bundles $O_{j,\ell}$ for $\ell \leq \ell'_j$. But this contradicts the fact that agent k is mapped to F_j in step z_1 , because agent k is mapped to O_{j,ℓ'_j} and no modification changes $M_j(k)$. □

Proof of Lemma 5.6. We want to show that in every step of the algorithm, for each remaining agent i , Inequality 4 holds.

$$\sum_{j \in S} w_{j,i} \cdot V_i(B_j) \geq \sum_{j < \ell_i} x_{i,j} \cdot v_{i,j}/2. \quad (4)$$

To prove this, we apply a sequence of exchanges between the bundles allocated to the agents in $\bigcup_{j < \ell_i} N_{i,j}$ and show that in every exchange, value of the expression on the left-hand side of Inequality (4) does not increase ⁵. Next, we show that after these exchanges, Inequality (4) holds, which means that the Inequality was held for the original allocation.

Let agent j be the agent in $N_{i,1}$ with the least influence on agent i (i.e., minimizes $w_{j,i}$). First, we allocate the bundles that belong to the other agents in $N_{i,1}$ to agent j and remove all the agents but agent j from $N_{i,1}$. Since agent j has the minimum weight among the agent in $N_{i,1}$, this operation does not increase the left-hand side of Inequality (4).

In addition, let agent j' be the agent with $w_{j',i} = x_{i,1}$. Since agent j' has the minimum weight among all the agents, $w_{j',i} \leq w_{j,i}$. Now, let B_j and $B_{j'}$ be the current bundles of agents j and j' . Note that if j' is not satisfied yet, $B_{j'} = \emptyset$. If $V_i(B_{j'}) < V_i(B_j)$, we swap the bundles of j and j' . This operation also does not increase the left-hand side of Inequality (4) since we have $w_{j',i} \leq w_{j,i}$. Finally, we exchange the set that agents j and j' belong to: we remove agent j from $N_{i,1}$, and add agent j' to $N_{i,1}$. In addition, if agent j' previously belonged to $N_{i,r}$ for some r , we add agent j to $N_{i,r}$. This exchange has no effect on the value of $\sum_{j \in S} w_{j,i} \cdot V_i(B_j)$. Furthermore, one can easily observe that after the exchange, condition (ii) holds.

We repeat the same procedure for $N_{i,2}, N_{i,3}, \dots, N_{i,\ell_i-1}$. After this sequence of exchanges, each $N_{i,j}$ contains one agent j' , where $w_{j',i} = x_{i,j}$. Furthermore, after the exchanges, the second condition for the validity of M_i holds and hence, the value of the items of agent j' for agent i is at least $v_{i,j}/2$. Therefore, total amount of externalities of the satisfied agents is at least $\sum_{j < \ell_i} x_{i,j} v_{i,j}/2$. ■

⁵Note that these exchanges are only to prove this lemma, and not in the algorithm.