Externalities and Fairness

Mohammad Ghodsi
Sharif University of Technology,
IPM - Institute for Research in Fundamental Sciences
Tehran, Iran
ghodsi@sharif.edu

Hamed Saleh
University of Maryland
College Park, Matyland, USA
hameelas@gmail.com

Masoud Seddighin
Sharif University of Technology
Tehran, Iran
mseddighin@ce.sharif.edu

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 the share 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. Inspired by the models in the context of network diffusion, 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. We further show that a $1/2$-approximation algorithm exists for this partitioning problem.

Next, we investigate on finding approximately optimal EMMS allocations, i.e., allocations that guarantee each 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.

CCS CONCEPTS

• **Theory of computation → Approximation algorithms analysis**; • **Applied computing → Economics**; • **Networks →** *Network resources allocation*; *Network economics*.

KEYWORDS

Fairness; maximin-share; approximation; externalities; networks

ACM Reference Format:
Mohammad Ghodsi, Hamed Saleh, and Masoud Seddighin. 2019. Externalities and Fairness. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313670

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313670

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 [28] 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 in cake-cutting, 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 instances[1].

For allocation of indivisible goods, Budish [10] introduced a new fairness criterion, namely *maximin-share*, that attracted a lot of attention in recent years [3, 6, 7, 14, 22, 27, 29]. 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 [27] 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. [14].

Our goal in this paper is to generalize maximin-share to the environment with externalities. Roughly speaking, externalities are the influences (costs or benefits) incurred by other parties. The consequences of various economic activities on third parties are studied by both economists and computer scientists. For resource allocation

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

problems, externalities imply that the bundle allocated to an agent may affect the utility of the other agents. These externalities can be either positive or negative. In this work, we assume that the externalities are **non-negative**, which is a common assumption in the literature [9, 17, 24].

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 exert externalities on his friends since they can borrow it.

We wish to take one step toward understanding the impact of externalities in the fair allocation of indivisible items. The messages of our paper can be summarized as follows. **First**, considering the externalities is important: the value of EMMS (the 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**, with respect to 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 techniques.

1.1 Our Contribution

We start by proposing a general model to capture the externalities in a fair allocation problem under the additive assumption. Although we present some of our results with respect to this general model, our focus 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 a cut-and-choose game. 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. We introduce a generalized form of the partition problem that includes many famous partition problems such as maximin, minimax, and leximin partitioning problems. This generalized problem is NP-hard due to a trivial reduction from the classic partition problem. In Section 3, we propose a $1/2$-approximation algorithm for computing EMMS_i (Theorem 3.2). 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 allocations, 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 $\beta/2$-EMMS_i (Theorem 4.2). 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 $\mathsf{EMMS}_i/2$. In every step of the algorithm, we choose an agent and allocate him a bundle with a value at least as his expectation value. Based on this allocation, we decrease the expectation value of the remaining agents. The process of updating the expectation values is a fairly complex process which we describe in Section 4.2. The analysis of the algorithm is technically involved and heavily exploits the structural properties of the general partitioning problem. Finally, the combination of our existential proof with the $1/2$-approximation algorithm for computing EMMS yields a polynomial time $\beta/4$-EMMS allocation algorithm.

1.2 Related Work

Maximin-share has received a lot of attention over the past few years [1–5, 8, 13–15, 18, 22, 27, 29]. The counter-example suggested by Procaccia and Wang [27] 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. [14] improve the approximation ratio to $3/4$. For the special case of 3 agents, Procaccia and Wang [27] prove that guaranteeing $3/4$ of every agent's maximin-share is always possible. Kurokawa et al. [22] 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 studies generalize maximin-share for different settings. For example, Farhadi et al. [13] generalize maximin-share to 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 [29] considers the case that the items must be allocated to groups of agents. Gourvès and Monnot [15] 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. Ghodsi et al. [14] and Barman et al. [7] consider maximin share guarantee for general valuation functions such as submodular, XOS, and subadditive set functions.

In recent years, considering externalities for different economic activities has received an increasing attention in computer science [4, 8, 9, 17, 19, 20, 23–25]. For example, Haghpanah et al. [17] study auction design in the presence of externalities. Pataki et al. [26] study the influences of the externalities on the Pareto optimal allocation of indivisible goods. Velez studies the fair allocation of indivisible goods and money with externalities [30]. In a more related work, Brânzei et al. [9] 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, Li et al. [24] 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 $[n]$

	b_1	b_2	b_3
1	1,5,2	4,6,1	7,1,0
2	3,8,5	1,8,7	4,6,9
3	1,5,3	6,5,7	5,4,1

Table 1: An instance in the general externalities model

denotes the set $\{1, 2, \ldots, n\}$. 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 item b, $V_{j,i}(b)$ reflects the utility that agent i receives by allocating b to agent j. In this model, there is no restriction on the value of $V_{j,i}(.)$. Total utility of agent i for an allocation is defined to be the sum of utilities he receives by every item, i.e.,

$$U_i = \sum_{b \in \mathcal{M}} V_{j_b,i}(b)$$

where j_b is the index of the agent whose b is allocated. For example, consider the instance with 3 items and 3 agents demonstrated in Figure 1. In this figure, for each agent i and item b_j, three values are given, where the k'th value shows the utility gained by agent i, if we allocate b_j to agent k. For this instance, if we allocate each item b_i to agent i, then we have:

$$\begin{aligned} U_1 &= 1 + 6 + 0 = 7, \\ U_2 &= 3 + 8 + 9 = 20, \\ U_3 &= 1 + 5 + 1 = 7. \end{aligned}$$

The main focus of the paper is on 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. In this model, each agent i has a valuation function V_i, where for each bundle S, $V_i(S)$ represents the happiness of agent i, if we allocate S to him. We assume that $V_i(.)$ is additive, i.e., for every bundle S,

$$V_i(S) = \sum_{b_j \in S} V_i(b_j).$$

In addition, 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 ratio 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 $V_i(\{b\}) \cdot w_{j,i}$. For convenience, we define $w_{i,i} = 1$ for every agent i and $w_{j,i} = 0$, for every agent j which is not connected via a directed edge to i.

For instance, consider the example illustrated in Figure 1. For the allocation that allocates every item b_i to agent i ($1 \le i \le 5$), the total utility of the agents would be

$$\begin{aligned} U_1 &= 5 + 5 \cdot 0.2 = 6, \\ U_2 &= 2, \\ U_3 &= 3 + 5 \cdot 0.2 + 2 \cdot 0.3 = 4.6, \\ U_4 &= 9 + 8 \cdot 0.1 + 2 \cdot 0.4 = 10.6, \\ U_5 &= 4 + 8 \cdot 0.25 + 7 \cdot 0.2 = 7.4. \end{aligned}$$

We call such a model the **network externalities** model.

Definition 2.1. For the network externalities model, we say agent i is β-self-reliant, if

$$\frac{1}{\sum_{1 \le j \le n} w_{j,i}} \ge \beta.$$

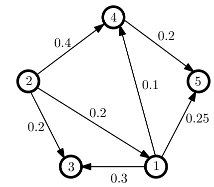
For example in Figure 1, agent 4 is $2/3$-self-reliant and agent 2 is 1-self-reliant. In real-world situations, we expect β to be a value close to 1. In other words, we expect an agent to be far more satisfied, when we allocate an item to him rather than allocating it to the other parties. However, being β-self-reliant for $\beta \simeq 1$ doesn't mean that we can ignore the externalities. We discuss more on this in Section 4.

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

As mentioned, we suppose that for every agent i, we have $w_{i,i} = 1$. As an example, for the graph in Figure 1, we have

$$\begin{aligned} x_1 &= [0, 0, 0, 0.2, 1], \\ x_2 &= [0, 0, 0, 0, 1], \\ x_3 &= [0, 0, 0.2, 0.3, 1], \\ x_4 &= [0, 0, 0.1, 0.4, 1], \\ x_5 &= [0, 0, 0.2, 0.25, 1]. \end{aligned}$$

We emphasize that throughout the paper, our assumption is that the externalities and the valuations are all non-negative.



	b_1	b_2	b_3	b_4	b_5
1	5	5	2	8	4
2	9	2	5	2	1
3	2	5	3	5	6
4	8	2	4	9	8
5	8	0	2	7	4

Figure 1: An instance in the network externalities model

2.2 Extended Maximin Share

In this paper, we introduce the extended maximin-share (EMMS) criterion. As mentioned, the maximin-share (MMS) notion was introduced by Budish [10] as a fairness criterion in the 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.

To extend maximin-share to the case of the agents with externalities, again we consider the worst-case scenario in a "cut and choose" game which incorporates externalities. 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). We define the *extended-maximin-share* of agent i as his maximin value in a "cut and choose" game with externalities.

Formally, let $P = \langle P_1, P_2, \ldots, P_n \rangle$ be a partition of $\mathcal{M}$ into n bundles with the following properties:

- For every i, we have $P_i \neq \emptyset$.
- For every i, j where $i \neq j$, we have $P_i \cap P_j = \emptyset$.
- $\cup_i P_i = \mathcal{M}$.

Furthermore, let $\mathcal{A} : P \to [n]$ be an allocation function that allocates every set P_i to agent $\mathcal{A}(P_i)$ (one set to each agent). 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:

$$U_i(\mathcal{A}) = \sum_j V_{\mathcal{A}_j, i}(P_j).$$

For example, consider the instance shown in Figure 2 and let

$$\mathcal{P} = \langle \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3 \rangle = \langle \{b_1, b_3\}, \{b_2, b_5\}, \{b_4\} \rangle$$

be a partition of the items into three bundles. One possible allocation is to allocate every bundle $\mathcal{P}_i$ to agent i. For this allocation (namely, $\mathcal{A}_{\mathcal{P}}$), we have

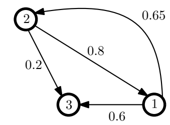
$$\begin{aligned}
U_1(\mathcal{A}_{\mathcal{P}}) &= (5+2) + (6+4) \cdot 0.8 = 15, \\
U_2(\mathcal{A}_{\mathcal{P}}) &= (9+5) \cdot 0.65 + (2+1) = 12.1, \\
U_3(\mathcal{A}_{\mathcal{P}}) &= (2+3) \cdot 0.6 + (0+6) \cdot 0.2 + 5 = 9.2.
\end{aligned}$$

The worst allocation of P for 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. For example, a worst allocation of $\mathcal{P}$ for agent 1 is to allocate $\mathcal{P}_3$ to agent 1, $\mathcal{P}_2$ to agent 3, and $\mathcal{P}_1$ to agent 2. The utility of agent 1 for this allocation is $3 + 7 \cdot 0.8 = 8.6$. Similarly, the best allocation of P is defined as:

$$\mathcal{B}_i(P) = \arg \max_{\mathcal{A} \in \Omega_P} U_i(\mathcal{A}).$$



	b_1	b_2	b_3	b_4	b_5
1	5	6	2	3	4
2	9	2	5	2	1
3	2	0	3	5	6

Figure 2: Another instance in the network externalities model

Again, the best allocation of $\mathcal{P}$ for agent 1 is to allocate $\mathcal{P}_2$ to agent 1, $\mathcal{P}_1$ to agent 2, and $\mathcal{P}_3$ to agent 3 which results in the utility of $10 + 7 \cdot 0.8 = 15.6$ for agent 1.

Finally, the extended-maximin-share of agent i, denoted by EMMS_i, is defined as:

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

where Π is the set of all partitions of $\mathcal{M}$ into n non-empty subsets. We also define the *optimal partition* of $\mathcal{M}$ for 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 $O_i = \langle O_{i,1}, O_{i,2}, \ldots, O_{i,n} \rangle$ are sorted by their non-increasing values for agent i, i.e., for all j, $V_i(O_{i,j}) \geq V_i(O_{i,j+1})$. Thus, we have [2]

$$\mathsf{EMMS}_i = \sum_{1 \leq j \leq n} V_i(O_{i,j}) \cdot x_{i,j}.$$

It can be observed that for agent 1 in Figure 2, we have

$$O_1 = \langle \{b_1, b_3\}, \{b_4, b_5\}, \{b_2\} \rangle$$

and

$$\begin{aligned}
\mathsf{EMMS}_1 &= U_1(\mathcal{W}_i(O_i)) \\
&= 7 \cdot 0.8 + 6 = 11.6.
\end{aligned}$$

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 for 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 $\alpha \cdot \mathsf{EMMS}_i$?

2.3 Average allocations

We also introduce another fairness notion for the extension of proportionality to the environments with externalities, namely *average-share*.

Definition 2.3 (average-share). The average value of item b for agent i, denoted by $\overline{V}_i(\{b\})$, is defined as $\sum_j V_{j,i}(\{b\})/n$. The average-share of agent i is $\overline{V}_i(\mathcal{M}) = \sum_{b \in \mathcal{M}} \overline{V}_i(\{b\})$. Furthermore, an allocation is said to be average, if the total utility of every agent from this allocation would be at least as his average-share.

[2] If it is not clear why this equation holds, see Section 3.

For the case of network externalities model, we have

$$\overline{V}_i(\mathcal{M}) = V_i(\mathcal{M})/n \cdot \sum_j w_{j,i}. \quad (1)$$

For the agents in Figure 2, we have

$$\overline{V}_1(\mathcal{M}) = (1.8/3) \cdot 20 = 12$$
$$\overline{V}_2(\mathcal{M}) = (1.65/3) \cdot 19 = 10.45$$
$$\overline{V}_3(\mathcal{M}) = (1.8/3) \cdot 16 = 9.6.$$

Average-share plays an important role in analyzing our algorithm for finding the (approximately) optimal partitions in Section 3. As we show in Section 2.4, this notion is stronger than extended maximin-share. However, no approximation of this notion can be satisfied even for very simple scenarios.

2.4 Basic Observations

In Section 2.1, we introduced two notions: average-share, and extended-maximin-share. For a better understanding of these notions, here we briefly compare them in the **general externalities** model. First, in Lemma 2.4, we prove that average share is stronger than extended maximin-share.

Lemma 2.4. *Average-share implies extended-maximin-share.*

Proof. Let $\mathbb{A} = \{A^1, A^2, \ldots, A^n\}$, 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 $\sum_j V_{j,i}(\mathcal{M})/n = \overline{V}_i(\mathcal{M})$ for agent i. As a result, $\mathsf{EMMS}_i = U_i(\mathcal{W}_i(O_i)) \leq \overline{V}_i(\mathcal{M})$. ◻

By a similar argument as in the proof of Lemma 2.4 we can show that for an arbitrary partition P,

$$\overline{V}_i(\mathcal{M}) \leq U_i(\mathcal{B}_i(P)).$$

Therefore, for any partition P we have

$$\mathsf{EMMS}_i \leq U_i(\mathcal{B}_i(P)).$$

In Lemma 2.5, we show that for $n = 2$, a *cut and choose* method guarantees EMMS_i to both the agents.

Lemma 2.5. *For two agents, the following two step algorithm yields a* 1-EMMS *allocation:*(i)*: Ask the first agent to partition the items into his optimal partition* O_1*, and* (ii)*: Ask the second agent to allocate* O_1 *(one bundle to each agent).*

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

$$U_1(\mathcal{B}_2(O_1)) \geq \mathsf{EMMS}_1.$$

◻

Note that there are instances with two agents such that no approximation of average-share can be guaranteed. For example, when there is only one item with value 1 to both the agents, and no externalities. We later establish another difference between these two notions by providing allocations that approximately guarantee extended maximin-share.

3 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 3.1 and Observation 1, we explore the hardness of these problems in the **general externalities** model. We then focus on the network externalities model and give a constant factor approximation algorithm for computing EMMS_i.

Lemma 3.1. *Given a partition* $P = \langle P_1, P_2, \ldots, P_n \rangle$ *of the items in* $\mathcal{M}$*, the worst allocation of* P *for agent* i *can be found in polynomial time.*

Proof. (sketch) 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 min-weight perfect matching in G. Classic network flow algorithms solve this problem in polynomial time [11]. ◻

Observation 1. *Since finding the maximin partition of a set of items is NP-hard [31], finding the optimal partition of m items and n agents with externalities is also NP-hard.*

Woeginger [31] also show 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 with externalities 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 remaining agent with the least effect on agent i. Hence,

$$U_i(\mathcal{W}_i(P)) = \sum_j x_{i,j} \cdot V_i(P_j). \quad (2)$$

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 classic 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 (2). Finding an optimal partition for a given vector x_i 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 $x_i = [1, 0, \ldots, 0]$. Likewise, minimax partition is a partition that minimizes the value of the maximum bundle, and it is the optimal partition when $x_i = [1, 1, \ldots, 1, 0]$. Another example is the leximin partition. A leximin partition first maximizes the minimum bundle, and subject to this

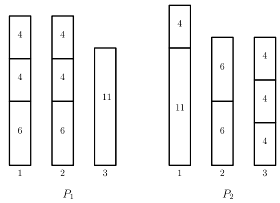


Figure 3: For $x_i = [1, 1, 0]$, minimax (P_1) is optimal, and for $x_i = [1, 0, 0]$, maximin (P_2) is optimal.

constraint, maximizes the second least valued bundle, and so on. Real-world applications of leximin allocations are recently studied by Kurokawa, Procaccia and Shah [21]. For a small enough ϵ, the optimal partition for vector $x_i = [1, \epsilon, \epsilon^2, ..., \epsilon^{n-1}]$ is a leximin partition. See Figure 3 for an illustrative example.

Since none of these partitioning schemes are always optimal, approximating either of them is not desirable. However, the well-known greedy algorithm LPT [3] provides a partition

$$L_i = \langle L_{i,1}, L_{i,2}, \ldots, L_{i,n} \rangle$$

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 has been previously established that the partition provided by LPT is a constant approximation for both maximin and minimax partitions [12, 16].

Theorem 3.2. *For the network externalities model, we have*

$$U_i(\mathcal{W}_i(L_i)) \geq \mathsf{EMMS}_i/2. \tag{3}$$

To prove Theorem 3.2, we label some of the items as *huge*. Let $\Delta = V_i(\mathcal{M})/n$, and define Huge items as those items whose values are at least Δ[4]. Denote the set of huge items for agent i by $\mathcal{H}_i$.

Lemma 3.3. *For an instance with no huge item, we have $V_i(L_{i,n}) \geq \Delta/2$.*

Proof. Consider $L_{i,1}$ (the most valuable bundle of L_i for agent i). Trivially, we have $V_i(L_{i,1}) \geq \Delta$, and since there is no huge item, $L_{i,1}$ contains at least two items. On the other hand, according to method of 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, $V_i(L_{i,n}) \geq V_i(L_{i,1})/2 \geq \Delta/2$. □

Since $L_{i,n}$ is the least valued bundle of L_i for agent i,

$$U_i(\mathcal{W}_i(L_i)) \geq \sum_i x_i V_i(L_{i,n}) \geq \sum_i x_i V_i(L_{i,1})/2 \geq \mathsf{EMMS}_i/2$$

Hence, when there is no huge item, regarding Lemma 3.3, Inequality (3) holds. Thus, to prove Theorem 3.2, it only suffices to consider the instances with huge items. Note that, when there are huge items in $\mathcal{M}$, $V_i(L_{i,n}) \geq \Delta/2$ does not necessarily hold. To deal with such situations, we consider some properties for O_i.

[3]Longest processing time

[4]Notice that Δ and the average share are not necessarily equal.

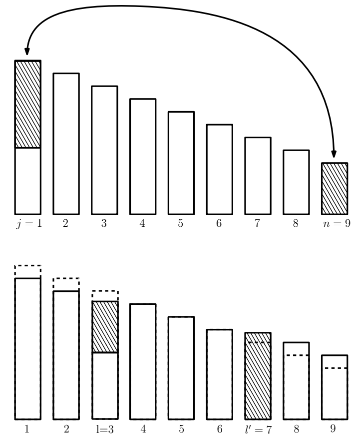


Figure 4: Switching the subsets

Definition 3.4. A partition P is *regular* 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 $V_i(P_n) = \min_j V_i(P_j)$).

Lemma 3.5. *For any partition P, there exists a regular partition P', such that $U_i(\mathcal{W}_i(P)) \leq U_i(\mathcal{W}_i(P'))$.*

Proof. If P is not regular, there exists an item b in some 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 non-increasing values for agent i), such that $j \leq l \leq l' \leq n$ (see Figure 4). 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 $(V_i(P_n), V_i(P_{n-1}), \ldots, V_i(P_1))$ increases lexicographically by each move, and hence we eventually end up with a regular partition P' after a finite number of modifications. □

Based on Lemma 3.5, in the rest of this paper we assume that the optimal partitions are regular. Furthermore, in Lemma 3.6 we show that L_i is also regular.

Lemma 3.6. *L_i is regular.*

Proof. For the sake of contradiction, let b be an item in bundle $L_{i,j}$ such that $V_i(L_{i,j}) > V_i(\{b\}) > V_i(L_{i,n})$. Since $V_i(L_{i,j}) > V_i(\{b\})$, $L_{i,j}$ contains at least one other item, say b'. Furthermore, since $V_i(\{b\}) > V_i(L_{i,n})$, after adding b to $L_{i,j}$ no other item can be added to $L_{i,j}$ (recall that in each step of LPT, we add an item to the least valued bundle). This means that b' is added to $L_{i,j}$ before b, which implies $V_i(b') \geq V_i(b)$. But this is a contradiction, because in the step that we add b to $L_{i,j}$, we have $V_i(L_{i,j}) \geq V_i(b') > V_i(L_{i,n})$, which means $L_{i,j}$ is not the minimum bundle in that step. □

In a regular partition P for agent i, any bundle P_j containing a huge item $b \in \mathcal{H}_i$ has no other item. Otherwise, since

$$V_i(P_j) > V_i(\{b\}) \geq \Delta,$$

and $\Delta > V_i(P_n)$, partition P is not regular. This fact about regular partitions (including L_i and O_i) allows us to deal with huge items. We are now ready to prove Theorem 3.2.

Proof of Theorem 3.2. We use induction on the number of agents. For $n = 1$, the statement is trivial. For $n > 1$, we consider two cases. First, if $\mathcal{M}$ contains no huge items, by Lemma 3.3, we have $V_i(L_{i,n}) \geq \Delta/2$. Thus,

$$\begin{aligned}
U_i(\mathcal{W}_i(\text{Ł}_i)) &= \sum_j V_i(L_{i,j}) \cdot x_{i,j} \\
&\geq \sum_j (\Delta/2) \cdot x_{i,j} \\
&= 1/2 \sum_j \Delta \cdot x_{i,j} \\
&= V_i(\mathcal{M})/2 \sum_j x_{i,j}/n \qquad (\Delta = V_i(\mathcal{M})/n) \\
&= \overline{V}_i(\mathcal{M})/2. \qquad \text{(Equation (1))}
\end{aligned}$$

By Lemma 2.4, we have $\overline{V}_i(\mathcal{M}) \geq \mathsf{EMMS}_i$ which implies

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

Therefore, it only remains to consider the case 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 regular). 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 Lemma 3.3 yields $V_i(L_{i,n}) \geq \Delta/2$.

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 $V_i(L_{i,n}) < \Delta/2$, 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 (3) holds. Now, adding huge items and their corresponding agents back, increases the utility of agent i by the same amount for both

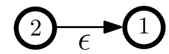


Figure 5: The gap between MMS_i and EMMS_i may be large, even with very small externalities

of the allocations. Thus,

$$U_i(\mathcal{W}_i(L_i)) \geq 1/2 \cdot U_i(\mathcal{W}'_i(O_i)) \geq 1/2 \cdot U_i(\mathcal{W}_i(O_i)).$$

□

4 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 to MMS_i.

Let M_i be the maximin partition of $\mathcal{M}$ for agent i. The maximin-share of agent i is equal to the value of the least valued bundle in M_i. Moreover, by definition, we have :

$$\mathsf{EMMS}_i = U_i(\mathcal{W}_i(O_i)) \geq U_i(\mathcal{W}_i(M_i)).$$

This, together with the fact that $U_i(\mathcal{W}_i(M_i)) \geq \mathsf{MMS}_i$ (recall that we have $w_{i,i} = 1$) implies that $\mathsf{EMMS}_i \geq \mathsf{MMS}_i$ always holds. Now, in Lemma 4.1 we show that the gap between EMMS_i and MMS_i could be unbounded even for the instances with 3 agents.

Lemma 4.1. *For any $c \geq 1$, there is an instance with 2 agents, where $\mathsf{EMMS}_1 > c \cdot \mathsf{MMS}_1$.*

Proof. Simply consider the influence graph depicted in Figure 5 and two items b_1 and b_2 such that $V_1(\{b_1\}) = 1$ and $V_1(\{b_2\}) = c/\epsilon$, where ϵ is a small constant less than $1/2$. For this instance, $\mathsf{EMMS}_1 = 1 + c$, and $\mathsf{MMS}_1 = 1$ which means

$$\mathsf{MMS}_1/\mathsf{EMMS}_1 < 1/(1 + c).$$

□

The proof of Lemma 4.1 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 4.2. We show that for the network externalities model when all the agents are α-self-reliant, an $\alpha/2$-EMMS allocation always exists. This completely separates extended maximin-share from average-share, since no approximation of the average-share can be guaranteed even for 1-self reliant agents.

Theorem 4.2. *Let $\mathbb{C}$ be an instance such that all the agents are α-self reliant. Then, $\mathbb{C}$ admits an $\alpha/2$-EMMS allocation.*

In the rest of this section, we prove Theorem 4.2 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* (BC) algorithm. It is worth to mention that despite some similarities, this method is fundamentally different from the previous allocation methods for guaranteeing MMS. The main difference is that, here, for each agent, we must keep track of the items allocated to the other parties and based on that, update the utility that each agent must receive to be satisfied. This makes the analysis much more complex.

4.1 Bundle Claiming Algorithm (BC)

In this section, we present the ideas and a general description of Bundle Claiming. 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 non-increasing values for agent i. Hence, for all $j < n$, we have $v_{i,j} \geq v_{i,j+1}$. We have $\mathsf{EMMS}_i = \sum_j x_{i,j} v_{i,j}$.

Let $X_i = \sum_j x_{i,j}$. Observation 2 provides an upper bound on the value of $\sum_{j \geq k} x_{i,j} v_{i,j}$.

Observation 2. *For every k, we have $\sum_{j \geq k} x_{i,j} v_{i,j} \leq v_{i,k} \cdot X_i$.*

For example, in an instance with $n = 6$, for $k = 4$, Observation 2 implies:

$$v_{i,4} \cdot X_i \geq v_{i,4} x_{i,4} + v_{i,5} x_{i,5} + v_{i,6} x_{i,6}.$$

Observation 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, $\sum_{j>k} x_{i,j} \leq X_i$.

Definition 4.3. 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 4.2. 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 $N_{i,1}, N_{i,2}, \ldots, N_{i,\ell_i-1}, N_{i,F}$, 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 hold. As we show in Section 4.2, during our algorithm, $\ell_i \leq n$ always holds for every agent i. We use 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 respect 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

ALGORITHM 1: Bundle Claiming algorithm

```
forall a_j ∈ N do
  | ℓ_j ← 1                          ▷ Initializing the expectation levels
end
while N ≠ ∅ do
  forall a_j ∈ N do
    Γ_j ← Minimum sized subset of M, s.t.
      V_j(Γ_j) ≥ 1/2 · v_{j,ℓ_j};
  end
  i ← arg min_i |Γ_i|;
  Allocate Γ_i to agent i ;
  N ← N \ i, M ← M \ Γ_i ;
  forall j ∈ N do
    N_{j,F} ← N_{j,F} ∪ {i}. ;
    while V_j(N_{j,F}) ≥ 1/2 · v_{j,l_j} do
      | Update(M_j)                   ▷ (see Section 4.2)
    end
  end
end
```

$$\begin{aligned}
&\sum_{k<\ell_i} v_{i,k} x_{i,k}/2 + v_{i,\ell_i}/2 \\
\geq &\sum_{k<\ell_i} v_{i,k} x_{i,k}/2 + \Big(\sum_{k\geq\ell_i} v_{i,k} x_{i,k}/X_i\Big)/2 && \text{(Observation 2)} \\
\geq &(1/2X_i) \sum_k v_{i,k} x_{i,k} && (X_i \geq 1) \\
\geq &\alpha/2 \sum_k v_{i,k} x_{i,k} \\
= &\alpha/2\mathsf{EMMS}_i. && (4)
\end{aligned}$$

Inequality (4) ensures that the items allocated to agent i satisfy him. Furthermore, we use 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 $\sum_{j\geq\ell_i} v_{i,j}$. Moreover, 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. In the next section, we show how to maintain the external-satisfaction property in the algorithm.

Note that, there are only two parts of the BC algorithm whose implementations in polynomial time are not trivial. One is finding a minimum-sized set B meeting the expectation of at least one agent, and the other is the operations we perform in order to maintain the external-satisfaction property. In Observation 3, we show that the first part can be implemented in polynomial time.

Observation 3. *The minimum-sized set that meets the expectation of at least one remaining agent can be found in polynomial time.*

Proof. 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 non-increasing 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. □

Furthermore, as we show in the next section, the operations we perform to preserve the external-satisfaction property can also be implemented in polynomial time.

Finally, using L_i [5] instead of O_i in BC results in an $\alpha/4$-EMMS allocation algorithm.

Corollary 4.4. *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.*

4.2 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 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 4.5. 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) $\quad \sum_{k \in N_{i,F}} V_i(B_k) < v_{i,\ell_i}/2$

During the entire 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 each step of the algorithm, we satisfy an agent i by a bundle B_i. 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 4.6.

Lemma 4.6. *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.*

Using Lemma 4.6 we can modify M_j and then update the mapping. Note that, after increasing ℓ_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 (Algorithm 2), you can find a pseudo-code for this process .

In the last part of this section, we prove Lemma 4.7 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 4.7.

Lemma 4.7. *Consider one step of the algorithm, and let agent i be an arbitrary remaining agent with $\ell_i > 1$. Then, we have*

$$\sum_{j \in \mathcal{S}} w_{j,i} \cdot V_i(B_j) \geq \sum_{j<\ell_i} x_{i,j} \cdot v_{i,j}/2. \tag{5}$$

Proof. We want to show that in every step of the algorithm, for each remaining agent i, Inequality (5) holds. 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 (5) does not increase [6]. Next, we show that after these exchanges, Inequality (5) 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 (influence) among the agent in $N_{i,1}$, this operation does not increase the left-hand side of Inequality (5).

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, we have $w_{j',i} \leq w_{j,i}$. Now, let B_j and $B_{j'}$ be the current bundles of agents j and j' (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 (5) 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

[5] Partitioning provided by LPT algorithm

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

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 $\sum_{j<\ell_i} x_{i,j} v_{i,j}/2$. □

4.3 Proof of Lemma 4.6.

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 also show in Lemma 4.8 that $|B_k| = 1$; but for now suppose that $|B_k| = 1$ and b is the only item in B_k.

Since O_j is regular[7], 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.

ALGORITHM 2: Update M_j

```
Resolve = 0
while Resolve == 0 do
    δ ← Minimum sized subset of N_{j,F}, s.t.
      ∑_{k∈δ} V_j(B_k) ≥ 1/2 · v_{j,ℓ_j}
    if ∑_{k∈δ} V_j(B_k) ≤ v_{j,ℓ_j} then
        N_{j,ℓ_j} = δ
        N_{j,F} = N_{j,f} \ δ
        ℓ_j ← ℓ_j + 1
        Resolve+ = 1                          ▷ Resolved
    else
        Let ℓ be an index s.t. O_{j,ℓ} = B_k, where δ = {k}.
        Swap δ (which is a subset of N_{j,F}) with N_{j,ℓ}. ▷ One
          step closer to resolve
    end
end
```

Lemma 4.8. *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 4.6, 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 is 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 1. *Either $v_{j,n} \geq v_{j,\ell'_j}/2$ or we have $|O_{j,\ell}| = 1$ for all $\ell \leq \ell'_j$.*

[7] Recall the regularity from Definition 3.4

Proof of Claim 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 3.5, moving s to bundle $O_{j,n}$ yields a new partition which is at least as good as O_j (See Figure 4). Hence, we can assume w.l.o.g. that $v_{j,n} \geq v_{j,\ell'_j}/2$ holds. □

Regarding Claim 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 te 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 $M_j(k)$. □

5 CONCLUSION

An exciting open direction is to find approximation allocation algorithms for the general externalities model, with no restriction on the value of $V_{j,i}(\{b_k\})$. Many issues complicate the study of the general model. For example, the approximation algorithm presented for computing the value of EMMS_i is no longer applicable to this model. A good starting point is to study the general model for the cases with a few number of agents, e.g., 3 or 4 agents.

For the network externalities model, one can think of improving the approximation ratio of the allocation. In particular, it would be interesting to propose an allocation algorithm with approximation factor independent of the self-reliance of the agents.

Another interesting open direction that might be of independent interest is to find a maximum value β, such that there exists a partition P of the items in which $\mathcal{W}_i(P)$ is a β-approximation of $\mathcal{W}_i(O_i)$ for every agents i. Note that if such a partition exists, any allocation of it to the agents is a β-EMMS allocation.

Finally, it is very impressive to present a *PTAS* for finding the optimal partition for an agent in the network externalities model.

REFERENCES

[1] Georgios Amanatidis, Georgios Birmpas, George Christodoulou, and Evangelos Markakis. 2017. Truthful allocation mechanisms without payments: Characterization and implications on fairness. In *Proceedings of the 2017 ACM Conference on Economics and Computation*. ACM, 545–562.

[2] Georgios Amanatidis, Georgios Birmpas, and Evangelos Markakis. 2016. On truthful mechanisms for maximin share allocations. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press, 31–37.

[3] Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. 2015. Approximation algorithms for computing maximin share allocations. In *International Colloquium on Automata, Languages, and Programming*. Springer, 39–51.

[4] Nima Anari, Shayan Ehsani, Mohammad Ghodsi, Nima Haghpanah, Nicole Immorlica, Hamid Mahini, and Vahab S Mirrokni. 2010. Equilibrium pricing with positive externalities. In *International Workshop on Internet and Network Economics*. Springer, 424–431.

[5] Haris Aziz, Gerhard Rauchecker, Guido Schryen, and Toby Walsh. 2017. Algorithms for Max-Min Share Fair Allocation of Indivisible Chores.. In *AAAI Conference on Artificial Intelligence*, Vol. 17. 335–341.

[6] Siddharth Barman, Arpita Biswas, Sanath Kumar Krishnamurthy, and Yadati Narahari. 2018. Groupwise maximin fair allocation of indivisible goods. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[7] Siddharth Barman and Sanath Kumar Krishna Murthy. 2017. Approximation algorithms for maximin fair division. In *Proceedings of the 2017 ACM Conference on Economics and Computation*. ACM, 647–664.

[8] Simina Brânzei, Tomasz Michalak, Talal Rahwan, Kate Larson, and Nicholas R Jennings. 2013. Matchings with externalities and attitudes. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 295–302.

[9] Simina Brânzei, Ariel D Procaccia, and Jie Zhang. 2013. Externalities in Cake Cutting.. In *International Joint Conference on Artificial Intelligence*. 55–61.

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

[11] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2009. *Introduction to algorithms*. MIT press.

[12] Bryan L Deuermeyer, Donald K Friesen, and Michael A Langston. 1982. Scheduling to maximize the minimum processor finish time in a multiprocessor system. *SIAM Journal on Algebraic Discrete Methods* 3, 2 (1982), 190–196.

[13] Alireza Farhadi, MohammadTaghi Hajiaghayi, Mohammad Ghodsi, Sebastien Lahaie, David Pennock, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. 2017. Fair Allocation of Indivisible Goods to Asymmetric Agents. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1535–1537.

[14] Mohammad Ghodsi, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. 2018. Fair Allocation of Indivisible Goods: Improvements and Generalizations. In *Proceedings of the 2018 ACM Conference on Economics and Computation*. ACM, 539–556.

[15] Laurent Gourvès and Jérôme Monnot. 2017. Approximate Maximin Share Allocations in Matroids. In *International Conference on Algorithms and Complexity*. Springer, 310–321.

[16] Ronald L. Graham. 1969. Bounds on multiprocessing timing anomalies. *SIAM journal on Applied Mathematics* 17, 2 (1969), 416–429.

[17] Nima Haghpanah, Nicole Immorlica, Vahab Mirrokni, and Kamesh Munagala. 2011. Optimal auctions with positive network externalities. In *Proceedings of the 12th ACM conference on Electronic commerce*. ACM, 11–20.

[18] Tobias Heinen, Nhan-Tam Nguyen, Trung Thanh Nguyen, and Jörg Rothe. 2018. Approximation and complexity of the optimization and existence problems for maximin share, proportional share, and minimax share allocation of indivisible goods. *Autonomous Agents and Multi-Agent Systems* (2018), 1–38.

[19] Miho Hong and Jaeok Park. 2018. Core and Top Trading Cycles in a Market with Indivisible Goods and Externalities. (2018).

[20] David Kempe and Mohammad Mahdian. 2008. A cascade model for externalities in sponsored search. In *International Workshop on Internet and Network Economics*. Springer, 585–596.

[21] David Kurokawa, Ariel D Procaccia, and Nisarg Shah. 2015. Leximin allocations in the real world. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*. ACM, 345–362.

[22] David Kurokawa, Ariel D Procaccia, and Junxing Wang. 2016. When can the maximin share guarantee be guaranteed?. In *AAAI Conference on Artificial Intelligence*, Vol. 16. 523–529.

[23] Renato Paes Leme, Vasilis Syrgkanis, and Éva Tardos. 2012. Sequential auctions and externalities. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 869–886.

[24] Minming Li, Jialin Zhang, and Qiang Zhang. 2015. Truthful cake cutting mechanisms with externalities: do not make them care for others too much!. In *International Joint Conference on Artificial Intelligence*. 589–595.

[25] Vahab S Mirrokni, Sebastien Roch, and Mukund Sundararajan. 2012. On Fixed-Price Marketing for Goods with Positive Network Externalities.. In *WINE*. Springer, 532–538.

[26] Eva Pataki, Andras Sagi, and Kabok Jozef. 2015. Externalities and the optimal allocation of economic resources. In *International Symposium on Intelligent Systems and Informatics*. IEEE, 185–188.

[27] Ariel D Procaccia and Junxing Wang. 2014. Fair enough: Guaranteeing approximate maximin shares. In *Proceedings of the fifteenth ACM conference on Economics and computation*. ACM, 675–692.

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

[29] Warut Suksompong. 2018. Approximate maximin shares for groups of agents. *Mathematical Social Sciences* 92 (2018), 40–47.

[30] Rodrigo A Velez. 2016. Fairness and externalities. *Theoretical Economics* 11, 1 (2016), 381–410.

[31] Gerhard J Woeginger. 1997. A polynomial-time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters* 20, 4 (1997), 149–154.