Noname manuscript No.
(will be inserted by the editor)

Maximin share Guarantee for Goods with Positive Externalities

Masoud Seddighin* · Hamed Saleh · Mohammad Ghodsi

Received: date / Accepted: date

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 with positive externalities. 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 approximate EMMS allocations, i.e., allocations that guarantee each agent a utility of at least a fraction of his extended-maximin-

Masoud Seddighin
School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran.

E-mail: seddighin.masood@gmail.com

Hamed Saleh
University of Maryland
E-mail: hameelas@gmail.com

Mohammad Ghodsi
Sharif University of Technology
School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran.
E-mail: ghodsi@sharif.edu

share. We show that under a natural assumption that the agents are α-self-reliant, an $\alpha/2$-EMMS allocation always exists. This, combined with the former result yields a polynomial-time $\alpha/4$-EMMS allocation algorithm.[1]

Keywords Fairness · Maximin-share · Externalities

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]. 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 for 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. For example, consider two agents and a single indivisible item. In any allocation scenario for this case, one of the agents receives nothing.

For allocation of indivisible goods, Budish [10] introduced a new fairness criterion, namely *maximin-share*, that attracted a lot of attention in recent years [14, 7, 6, 29, 23, 3, 27]. 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

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

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 later improved to $3/4$ by Ghodsi et al. [14].

Our goal in this paper is to generalize the notion of maximin-share to the environment with externalities. Roughly speaking, externalities are the

[1] A version of this work is accepted in TheWebConf (WWW) 2019.

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 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 [17,9,25].

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 useful. **Second**, with respect to our model and fairness notion, we can approximately maintain fairness in the environment with positive externalities. In Section 1.1, we give a more detailed explanation 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 models in the context of network diffusion.

We suggest the *extended-maximin-share* (EMMS) notion 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 5, we propose a 1/2-approximation algorithm for computing EMMS_i (Theorem 1). In fact, we show that the LPT method, which is a famous greedy algorithm in 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 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 6.2. The analysis of the algorithm is technically involved and heavily exploits the structural properties of the general partitioning problem. Finally, a combination of our existential proof with the 1/2-approximation algorithm for computing EMMS yields a polynomial time $\beta/4$-EMMS allocation algorithm.

2 Related Work

Maximin-share has received a lot of attention over the past few years [14, 19, 15, 23, 13, 5, 1, 29, 2, 3, 27, 8, 4]. 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. This factor is later improved to 7/8 by Amanatidis et al. [3] and to 8/9 by Gourvès and Monnot [15]. Kurokawa et al. [23] 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. Li and Vetta consider maximin share for the case where the items form a hereditary set system and propose an algorithm that allocates every agent a share with value at least 11/30-MMS.

In recent years, considering externalities for different economic activities has received an increasing attention in computer science [21,17,9,25,24,4,26, 20]. For example, Haghpanah et al. [17] study auction design in the presence of externalities. In their setting, a bidder's value for an outcome is a fixed private value times a certain submodular function of the share of his friends. Indeed, our model can be considered as a special case of their model, where the externalities is a linear function of the allocation of ones friends. Velez studies the fair allocation of indivisible goods and money with externalities [30]. In their setting, there are n agents and n items, each agent must receive one item and an amount of money. They consider two types of externalities. In their first setting, an agent loses utility when his perception of the value of the share of others deviates from the value of his own share. In another setting, they consider the case that externalities are a linear function of the money allocated to the other agents.

In a more related work, Brânzei et al. [9] consider positive externalities in the cake cutting problem. They introduce a model for cake cutting with externalities and generalize classic fairness criteria to the case with positive externalities. In their model each agent i has n value density functions $v_{i,j}(x)$ for $1 \leq j \leq n$, where $v_i, j(x)$ determines the value that i receives when x is allocated to agent j.

Following this work, Li et al. [25] study truthful and fair methods for allocating a divisible resource with positive externalities. In their model, the utility of an agent for the allocation is his valuation for his share plus certain ratio of other agents' valuations for their shares. In contrast to our model, in their model the utility of an agent depends on the valuations reported by other parties.

3 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 $\{1, 2, \ldots, n\}$. We introduce our model in Section 3.1 and our fairness criterion in Section 3.2.

3.1 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, $V_{i,j}(b)$ reflects the utility that agent i receives by allocating b to agent j. In this model, there is no restriction on the value of $V_{i,j}(\cdot)$, except that for every item b, $V_{i,j}(b) \geq 0$.

	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

Total utility of agent i for an allocation is defined to be the sum of utilities he receives by every item, i.e.,

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

where j_b is the index of the agent whose item b is allocated. For example, consider the instance with 3 items and 3 agents demonstrated in Table 1. In this table, 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, we have:

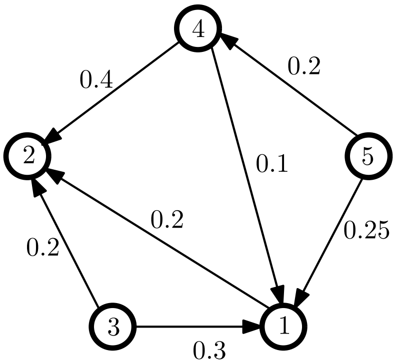
$$U_1 = 1 + 6 + 0 = 7,$$
$$U_2 = 3 + 8 + 9 = 20,$$
$$U_3 = 1 + 5 + 1 = 7.$$

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 $V_i(\cdot)$ 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 $\overrightarrow{(i,j)}$, denoted by $w_{i,j}$, 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 $V_i(b) \cdot w_{i,j}$. 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. It is natural to assume $w_{i,j} \leq 1$ for every $i \neq j$, which means every agent i derives less utility from some other agent receiving a good than he himself.

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



	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

Fig. 1 An instance in the network externalities model

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 1 For the network externalities model, we say agent i is α-self-reliant, if

$$\frac{1}{\sum_{1 \leq j \leq n} w_{i,j}} \geq \alpha \tag{1}$$

It's worth pointing out that the 1 in the numerator of Equation (1) refers to $w_{i,i}$. As an example, in Figure 1 agent 4 is 2/3-self-reliant and agent 2 is 1-self-reliant. In real-world situations, we expect α 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 α-self-reliant for α very close to 1 (but not equal) doesn't mean that we can ignore the externalities. We discuss more on this in Section 6.

Definition 2 For every agent i, we define the *influence vector* of agent i, denoted by $\mathbf{x}_{i\uparrow} = [\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,n}]$ as the vector representing the influences of the agents on agent i in the influence graph, in a non-decreasing order (we use symbol $\uparrow$ in the subscription to remind the non-decreasing order of influence vectors).

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

$$\begin{aligned}
\mathbf{x}_{1\uparrow} &= [0, 0, 0, 0.2, 1],\\
\mathbf{x}_{2\uparrow} &= [0, 0, 0, 0, 1],\\
\mathbf{x}_{3\uparrow} &= [0, 0, 0.2, 0.3, 1],\\
\mathbf{x}_{4\uparrow} &= [0, 0, 0.1, 0.4, 1],\\
\mathbf{x}_{5\uparrow} &= [0, 0, 0.2, 0.25, 1].
\end{aligned}$$

3.2 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. In contrast to proportionality and envy-freeness, guaranteeing a constant fraction of the maximin-share to all the agents is always possible [27, 14].

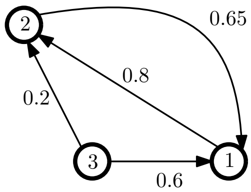
To extend maximin-share to the case of the agents with externalities, again we consider the worst-case scenario in a "cut and choose" game which incorporates externalities. Suppose that an agent divides the items into n bundles, and other agents somehow distribute these bundles (one bundle to each agent). 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.

Every allocation of items to the agents is defined as a pair (P, σ), where σ is a permutation of $[n]$, and $P = \langle P_1, P_2, \ldots, P_n \rangle$ is a partition of $\mathcal{M}$ into n bundles with the following properties:

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

For an allocation pair (P, σ), we allocate bundle P_i to σ_i. Therefore, the utility of agent i for allocation pair (P, σ) is

$$U_i(P, \sigma) = \sum_{1 \leq j \leq n} V_{i,\sigma(j)}(P_j).$$



	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

Fig. 2 Another instance in the network externalities model

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

$$P = \langle P_1, P_2, P_3 \rangle = \langle \{b_1, b_3\}, \{b_2, b_5\}, \{b_4\} \rangle$$

be a partition of the items into three bundles. For $\sigma = 1, 2, 3$, we have

$$U_1(P, \sigma) = (5 + 2) + (6 + 4) \cdot 0.8 = 15,$$
$$U_2(P, \sigma) = (9 + 5) \cdot 0.65 + (2 + 1) = 12.1,$$
$$U_3(P, \sigma) = (2 + 3) \cdot 0.6 + (0 + 6) \cdot 0.2 + 5 = 9.2.$$

The worst permutation of P for agent i, denoted by $\omega_i(P)$, is the permutation of $[n]$ that minimizes the utility of agent i:

$$\omega_i(P) = \arg\min_{\sigma \in \Sigma} \quad U_i(P, \sigma),$$

where Σ is the set of all $n!$ different permutations of $[n]$. For example, in Figure 2, the worst permutation of $P = \langle \{b_1, b_3\}, \{b_2, b_5\}, \{b_4\} \rangle$ for agent 1 is $[2, 3, 1]$ which allocates P_3 to agent 1, P_2 to agent 3, and P_1 to agent 2. The utility of agent 1 for this permutation is $3 + 7 \cdot 0.8 = 8.6$. Similarly, the best permutation of P is defined as:

$$\beta_i(P) = \arg\max_{\sigma \in \Sigma} \quad U_i(P, \sigma).$$

Again, the best allocation of P for agent 1 is $[2, 1, 3]$, which allocates P_2 to agent 1, P_1 to agent 2, and P_3 to agent 3 and results in the utility of $10 + 7 \cdot 0.8 = 15.6$ for agent 1. When the agent and the partition is clear from the context, we refer the best and worst permutation by β and ω, respectively.For example, we use $U_i(P, \omega)$ instead of $U_i(P, \omega_i(P))$.

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

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

where Π is the set of all partitions of $\mathcal{M}$ into n non-empty subsets. We also define the *Optimal* EMMS *partition* of $\mathcal{M}$ for agent i, denoted by $O_{i\downarrow}$, as the partition that determines the value of EMMS_i,

$$O_{i\downarrow} = \arg\max_{P\in\Pi} \quad U_i(P,\omega).$$

We assume that the bundles in $O_{i\downarrow} = \langle O_{i,1}, O_{i,2}, \ldots, O_{i,n}\rangle$ are sorted by their non-increasing values for agent i, i.e., for all j, $V_i(O_{i,j}) \geq V_i(O_{i,j+1})$ (we use $\downarrow$ in the subscription to remind the non-increasing order of the bundles). We also suppose that $\mathbf{v}_{i\downarrow} = [\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \ldots, \mathbf{v}_{i,n}]$ is a vector representing values of the bundles in the optimal EMMS partition of agent i, i.e., $\mathbf{v}_{i,j} = V_i(O_{i,j})$. Again, we note that the values in $\mathbf{v}_{i\downarrow}$ are non-increasing. Following **Rearrangement inequality** [18] [2] we have:

$$\mathsf{EMMS}_i = \sum_{1\leq j\leq n} \mathbf{v}_{i,j} \cdot \mathbf{x}_{i,j}.$$

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

$$O_{1\downarrow} = \langle\{b_1, b_3\}, \{b_4, b_5\}, \{b_2\}\rangle$$

which means $\mathbf{v}_{1\downarrow} = [7, 7, 6]$. Since $\mathbf{x}_{i\uparrow} = [0, 0.8, 1]$, we have

$$\begin{aligned}\mathsf{EMMS}_1 &= U_1(O_{i\downarrow}, \omega)\\ &= 7\cdot 0 + 7\cdot 0.8 + 1\cdot 6 = 11.6.\end{aligned}$$

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

3.3 Average allocations

We also introduce average-share which can be seen as an extension of proportionality to the environments with externalities. Average-share plays an important role in analyzing our algorithm for finding the (approximately) optimal EMMS partitions in Section 5.

Definition 3 (average-share) The average value of item b for agent i, denoted by $\overline{V}_i(b)$, is defined as

$$\sum_j V_{i,j}(b)/n.$$

[2] A simple form of this inequality states that for every choice of real numbers $x_1 \leq x_2 \leq \ldots \leq x_n$ and $y_1 \geq y_2 \geq \ldots \geq y_n$ and every permutation σ of $[n]$, we have:

$$x_1y_1 + x_2y_2 + \ldots + x_ny_n \leq x_{\sigma(1)}y_1 + x_{\sigma(2)}y_2 + \ldots + x_{\sigma_n}y_n.$$

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.

For the case of network externalities we have

$$\overline{V}_i(\mathcal{M}) = (V_i(\mathcal{M})/n) \cdot \sum_j w_{i,j}. \quad (2)$$

As an example, for the agents in Figure 2 we have

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

As we show in Section 4, this notion is stronger than extended maximin-share. However, no approximation of this notion can be satisfied even for very simple scenarios.

Remark 1 We emphasize our assumption that the externalities and the valuations are all non-negative. Even though the definition of EMMS and average-share and our method for approximating EMMS remain valid for negative externalities, our α-EMMS allocation algorithm essentially relies on positive externalities.

4 Basic Observations

In Section 3, 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 1, we prove that average-share is stronger than extended-maximin-share.

Lemma 1 *For every agent i we have $\mathsf{EMMS}_i \leq \overline{V}_i(\mathcal{M})$.*

Proof. Let $\sigma_1, \sigma_2, \ldots, \sigma_n$ be n different permutations, where

$$\sigma_k(j) = ((j + k) \bmod n) + 1.$$

Since in the total of these n allocations, each item is allocated to each agent once, we have

$$\sum_{1 \leq j \leq n} U_i(O_{i\downarrow}, \sigma_j) = \sum_{1 \leq j \leq n} V_{i,j}(\mathcal{M}).$$

Thus, the worst allocation among these n allocations has a utility of at most

$$\sum_{1 \le j \le n} V_{i,j}(\mathcal{M})/n = \overline{V}_i(\mathcal{M})$$

for agent i. As a result, $\mathsf{EMMS}_i = U_i(O_{i\downarrow}, \omega) \le \overline{V}_i(\mathcal{M})$.

□

By a similar argument as in the proof of Lemma 1 we can show that for an arbitrary partition P,

$$\overline{V}_i(\mathcal{M}) \le U_i(P, \beta).$$

Therefore, for any partition P we have

$$\mathsf{EMMS}_i \le U_i(P, \beta).$$

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

Lemma 2 *For two agents, the following two step algorithm yields a* 1-EMMS *allocation:*

- *Ask the first agent to partition the items into his optimal* EMMS *partition* $O_{1\downarrow}$.
- *Ask the second agent to allocate* $O_{1\downarrow}$ *(one bundle to each agent).*

Proof. We know $U_2(O_{1\downarrow}, \beta_2(O_{1\downarrow})) \ge \mathsf{EMMS}_2$. Furthermore, since permutation $\omega_1(O_{1\downarrow})$ determines the value of EMMS_1, we have:

$$U_1(O_{1\downarrow}, \beta_2(O_{1\downarrow})) \ge \mathsf{EMMS}_1.$$

□

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 an allocation algorithm that approximately guarantees extended maximin-share.

In Table 4, you can find a summary of the notations and definitions provided in this section.

Notation	Description
$P_{i\downarrow}$	Bundles in partition P are ordered by their non-increasing values for agent i
$U_i(P, \sigma)$	Utility of agent i for allocation (P, σ)
$\beta_i(P)$	Best allocation of partition P for agent i
$\omega_i(P)$	Worst allocation of partition P for agent i
$O_{i\downarrow} = \langle O_{i,1}, O_{i,2}, \ldots, O_{i,n} \rangle$	Optimal partition of agent i
$\mathbf{x}_{i\uparrow} = [\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,n}]$	Influence vector of agent i (non-dec)
$\mathbf{v}_{i\downarrow} = [\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \ldots, \mathbf{v}_{i,n}]$	value vector of O_i (non-inc)
$\mathsf{EMMS}_i = U_i(O_{i\downarrow}, \omega) = \mathbf{v}_{i\downarrow} \cdot \mathbf{x}_{i\uparrow}$	Extended-maximin-share value of agent i
$\overline{V}_i(\mathcal{M}) = (V_i(\mathcal{M})/n) \cdot w_{i,j}$	Average-share of agent i

Table 2 List of the notations mentioned in Section 3

5 Computing EMMS

In this section, we study the problem of computing EMMS_i and $O_{i\downarrow}$. 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 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 *Given a partition $P = \langle P_1, P_2, \ldots, P_n \rangle$ of the items in $\mathcal{M}$, the worst permutation of P for agent i can be found in polynomial time.*

Proof. 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 $V_{i,j}(P_k)$ between every pair $x_k \in X$ and $y_j \in Y$. Finding $\omega_i(P)$ 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* [3] *of a set of items (without externalities) is NP-hard [31], finding the optimal* EMMS *partition of m items and n agents with externalities is also NP-hard.*

Woeginger [31] also shows 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 EMMS partition for an agent with externalities has not been studied before.

[3] A partition of items into n bundles that maximizes the value of the minimum bundle.

In the case of *network externalities*, our model is easier to deal with. Consider a partition $P_{i\downarrow} = \langle P_{i,1}, P_{i,2}, \dots, P_{i,n} \rangle$ whose bundles are sorted in a non-increasing order of their values for agent i. Based on Rearrangement inequality, finding the worst allocation $\omega_i(P_{i\downarrow})$ 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,

$$U_i(P_{i\downarrow}, \omega) = \sum_{1 \le j \le n} \mathbf{x}_{i,j} \cdot V_i(P_{i,j}). \tag{3}$$

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.

Apart from allocating bundles, partitioning the items is another challenge to overcome. By definition, an optimal partition is a partition that maximizes Equation (3). Finding an optimal EMMS partition for a given influence vector $\mathbf{x}_{i\uparrow}$ 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 $\mathbf{x}_{i\uparrow} = [0, \dots, 0, 1]$. Likewise, minimax partition is a partition that minimizes the value of the maximum bundle, and it is the optimal EMMS partition when $\mathbf{x}_{i\uparrow} = [0, 1, \dots, 1, 1]$ (see Figure 3 for an illustrative example). 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 [22]. In Lemma 4 we show that for a small enough ϵ, the optimal EMMS partition for vector $\mathbf{x}_{i\uparrow} = [\epsilon^{n-1}, ..., \epsilon^2, \epsilon, 1]$ is the leximin partition.

Lemma 4 *Let*

$$\delta = \min_{S,T \subseteq \mathcal{M}, V_i(S) \neq V_i(T)} |V_i(S) - V_i(T)|$$

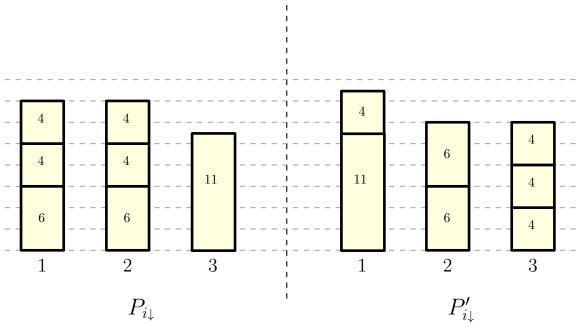
be the smallest positive difference between the values of any two bundles for agent i and let

$$\epsilon \le \delta / V_i(\mathcal{M})$$

be a positive constant. The optimal EMMS *partition for* $\mathbf{x}_{i\uparrow} = [\epsilon^{n-1}, ..., \epsilon^2, \epsilon, 1]$ *is Leximin.*

Proof. Let $\mathsf{LEX}_{i\downarrow} = \langle \mathsf{LEX}_{i,1}, \mathsf{LEX}_{i,2}, \dots, \mathsf{LEX}_{i,n} \rangle$ be the leximin partition of agent i, and let $P_{i\downarrow} = \langle P_{i,1}, P_{i,2}, \dots, P_{i,n} \rangle$ be any other partition. We show

$$\sum_{1 \le j \le n} \mathbf{x}_{i,j} \cdot V_i(\mathsf{LEX}_{i,j}) \ge \sum_{1 \le j \le n} \mathbf{x}_{i,j} \cdot V_i(P_{i,j}). \tag{4}$$



$\mathbf{x}_{i\downarrow} = [1,1,0]$ $U_i = 1 \cdot 14 + 1 \cdot 14 + 0 \cdot 11 = 28$ $U_i = 1 \cdot 15 + 1 \cdot 12 + 0 \cdot 12 = 27$

$\mathbf{x}_{i\downarrow} = [1,0,0]$ $U_i = 1 \cdot 14 + 0 \cdot 14 + 0 \cdot 11 = 14$ $U_i = 1 \cdot 15 + 0 \cdot 12 + 0 \cdot 12 = 15$

Fig. 3 Two partitions $P_{i\downarrow}$ and $P'_{i\downarrow}$ and their utilities for two influence vectors $\mathbf{x}_{i\uparrow} = [1,1,0]$ and $\mathbf{x}_{i\uparrow} = [1,0,0]$. For $\mathbf{x}_{i\uparrow} = [1,1,0]$, minimax ($P_{i\downarrow}$) is optimal, and for $\mathbf{x}_{i\uparrow} = [1,0,0]$, maximin ($P'_{i\downarrow}$) is optimal.

Let k be the largest index, such that $V_i(\mathsf{LEX}_{i,k}) \neq V_i(P_{i,k})$. By definition, $V_i(\mathsf{LEX}_{i,k}) \geq V_i(P_{i,k})$ holds. We have

$$\begin{aligned}
\sum_{1 \leq j \leq n} \mathbf{x}_{i,j} \cdot V_i(\mathsf{LEX}_{i,j}) &\geq \sum_{k \leq j \leq n} \mathbf{x}_{i,j} \cdot V_i(\mathsf{LEX}_{i,j}) \\
&= \mathbf{x}_{i,k} \cdot (V_i(\mathsf{LEX}_{i,k}) - V_i(P_{i,k})) + \sum_{k \leq j \leq n} \mathbf{x}_{i,j} \cdot V_i(P_{i,j}) \\
&\geq \mathbf{x}_{i,k} \cdot \delta + \sum_{k \leq j \leq n} \mathbf{x}_{i,j} \cdot V_i(P_{i,j})
\end{aligned}$$

Thus, in order for Inequality (4) to hold, it only suffices to choose ϵ such that

$$\mathbf{x}_{i,k} \cdot \delta \geq \sum_{1 \leq j \leq k-1} \mathbf{x}_{i,j} \cdot V_i(P_{i,j}). \tag{5}$$

A trivial upper-bound on the right-hand side of Inequality (5) is $\epsilon^{n-k+1} \cdot V_i(\mathcal{M})$. Since $\mathbf{x}_{i,k} = \epsilon^{n-k}$, if we choose ϵ, such that $1/\epsilon \geq V_i(\mathcal{M})/\delta$, Inequality (4) holds. Thus, it only suffices to choose $\epsilon < \delta/V_i(\mathcal{M})$.

□

In the rest of this section, we prove that LPT,[4] which is a well-known greedy method in job scheduling, provides a partition

$$\mathsf{LPT}_{i\downarrow} = \langle \mathsf{LPT}_{i,1}, \mathsf{LPT}_{i,2}, \ldots, \mathsf{LPT}_{i,n} \rangle$$

for agent i, such that $U_i(\mathsf{LPT}_{i\downarrow}, \omega)$ is a constant approximation of EMMS_i. This algorithm starts with n empty bundles and iteratively puts the most

[4] Longest processing time

Notation	Description
μ	$V_i(\mathcal{M})/n$
$\mathcal{H}_P$	set of bundles containing huge items (items with value at least μ) in P for agent i
$\mathsf{LPT}_{i\downarrow}$	Partition returned by LPT algorithm

Table 3 List of the notations mentioned in Section 3.

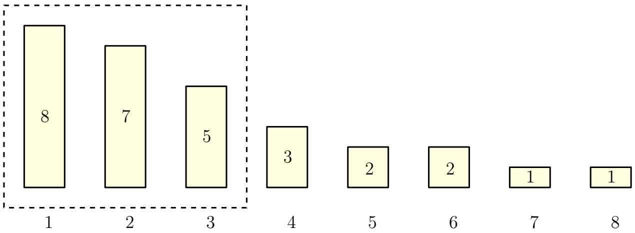


Fig. 4 For the example in this figure, we have $\mu = V_i(\mathcal{M})/n = 29/8 = 3.625$. Therefore, the first three items are huge.

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 [16,12].

Theorem 1 *For the network externalities model, we have*

$$U_i(LPT_{i\downarrow}, \omega) \geq \mathsf{EMMS}_i/2. \tag{6}$$

In the rest of this section, we prove Theorem 1. To prove Theorem 1, we label some of the items as *huge*. Let $\mu = V_i(\mathcal{M})/n$, and define huge items as those items whose values for agent i are at least μ. For example, the first three items in Figure 4 are huge. In Table 3 you can find a list of the notations that are frequently used in this section.

First, in Lemma 5 and Observation 6, we show how to handle the instances with no huge item.

Lemma 5 *For an instance with no huge item, we have*

$$V_i(LPT_{i,n}) \geq V_i(LPT_{i,1})/2 \geq \mu/2.$$

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

$$V_i(\mathsf{LPT}_{i,n}) \geq V_i(\mathsf{LPT}_{i,1})/2 \geq \mu/2. \tag{7}$$

□

Now, in Lemma 6 we show that when $V_i(\mathsf{LPT}_{i,n}) \geq \mu/2$, $\mathsf{LPT}_{i\downarrow}$ is a 2-approximation of $O_{i\downarrow}$.

Lemma 6 *If $V_i(LPT_{i,n}) \geq \mu/2$, we have*

$$U_i(LPT_{i\downarrow}, \omega) \geq \mathsf{EMMS}_i/2.$$

Proof. We have

$$\begin{aligned}
U_i(\mathsf{LPT}_{i\downarrow}, \omega) &= \sum_{1 \leq j \leq n} \mathbf{x}_{i,j} \cdot V_i(\mathsf{LPT}_{i,j}) \\
&\geq \sum_{1 \leq j \leq n} \mathbf{x}_{i,j} \cdot V_i(\mathsf{LPT}_{i,n}) \\
&\geq \sum_{1 \leq j \leq n} \mathbf{x}_{i,j} \cdot \mu/2 \\
&\geq (V_i(\mathcal{M})/2) \cdot \sum_{1 \leq j \leq n} \mathbf{x}_{i,j}/n && (\mu = V_i(\mathcal{M})/n) \\
&\geq \overline{V}_i(\mathcal{M})/2 && \text{Equation (2)} \\
&\geq \mathsf{EMMS}_i/2. && \text{Lemma (1)}
\end{aligned}$$

□

An immediate corollary of Lemmas 5 and 6 is Corollary 1, which states that when there is no huge item, $\mathsf{LPT}_{i\downarrow}$ is a 2-approximation of $O_{i\downarrow}$.

Corollary 1 *When there is no huge item, we have*

$$U_i(LPT_{i\downarrow}, \omega) \geq \mathsf{EMMS}_i/2.$$

Thus, to prove Theorem 1, it only suffices to consider the instances with huge items. Note that, when there are huge items in $\mathcal{M}$, $V_i(\mathsf{LPT}_{i,n}) \geq \mu/2$ does not necessarily hold. To deal with such situations, we consider some properties for $O_{i\downarrow}$. In Definition 4, we introduce regular partitions. Roughly speaking, a partition $P_{i\downarrow}$ is regular, if every bundle with more than one item admits no valuable item (i.e., item with value more than the value of the least attractive bundle of $P_{i\downarrow}$). In other words, each valuable item belongs to a single-item bundle. This property greatly helps up deal with huge items.

Definition 4 A partition $P_{i\downarrow}$ is *regular* for agent i, if no item b in some bundle $P_{i,j}$ exists, such that

$$V_i(P_{i,j}) > V_i(b) > V_i(P_{i,n}).$$

For example, in Figure 5, partition $P_{i\downarrow}$ is not regular since the total value of the items in the third bundle is less than the item with value 5 in the first bundle. In contrast to $P_{i\downarrow}$, it is easy to verify that partition $P'_{i\downarrow}$ is regular.

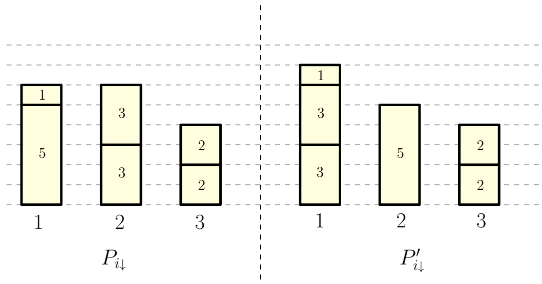


Fig. 5 Partition P_1 is not regular, since the bundle containing item with value 5, which is more than the total value of the third bundle, is not in a single-item bundle.

Lemma 7 *For any partition $P_{i\downarrow}$, there exists a regular partition $P'_{i\downarrow}$, such that $U_i(P_{i\downarrow}, \omega) \le U_i(P'_{i\downarrow}, \omega)$.*

Proof. If $P_{i\downarrow}$ is not regular, there exists an item b in some bundle $P_{i,j}$, such that

$$V_i(P_{i,j}) > V_i(b) > V_i(P_{i,n}).$$

We modify $P_{i\downarrow}$ as follows: we remove $P_{i,j}$ and $P_{i,n}$ from $P_{i\downarrow}$ and add two new bundles $A = \{b\}$ and $B = P_{i,j} \cup P_{i,n} \setminus \{b\}$ to $P_{i\downarrow}$. Let $P'_{i\downarrow}$ be the partition after the modification and let l and l' be the indices of the bundles correspond to A and B (note that the bundles are rearranged by their non-increasing values for agent i), such that $j \le l \le l' \le n$. We have:

$$V_i(P_{i,j}) > V_i(P'_{i,l}) \ge V_i(P'_{i,l'}) > V_i(P_{i,n}).$$

For example, consider the instance in Figure 6. In this figure, partition $P_{i\downarrow}$ is not regular since $P_{i,2}$ admits an item with value more than the value of the entire bundle $P_{i,10}$. After performing the above operation on $P_{i\downarrow}$ we obtain partition $P'_{i\downarrow}$. In this partition, all the partitions except $P'_{i,4}$ and $P'_{i,8}$ are the same as in $P_{i\downarrow}$; only their position might have changed.

Define Δ_j to be the difference between the value of $P'_{i,j}$ and $P_{i,j}$, i.e.,

$$\Delta_j = V_i(P'_{i,j}) - V_i(P_{i,j}).$$

Since the set of items in $P_{i\downarrow}$ and $P'_{i\downarrow}$ are the same, we have

$$\sum_{1 \le k \le n} \Delta_k = 0. \tag{8}$$

In addition, for $k < j$ and $l < k < l'$ we have $\Delta_k = 0$, for $j \le k \le l$ we have $\Delta_k \le 0$ and for $l' < k \le n$ we have $\Delta_k \ge 0$. For example, in Figure 6, we have

$$\begin{aligned} \Delta_1 = \Delta_5 = \Delta_6 = \Delta_7 = 0 \\ \Delta_2, \Delta_3, \Delta_4 \le 0 \\ \Delta_8, \Delta_9, \Delta_{10} \ge 0 \end{aligned}$$

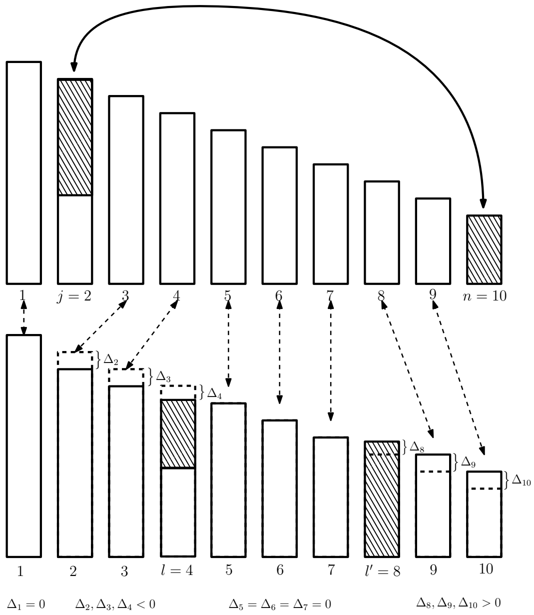


Fig. 6 Switching the subsets

Now, we have

$$\begin{aligned}
\sum_{1\le k\le n} V_i(P'_{i,k})\cdot \mathbf{x}_{i,k} - \sum_{1\le k\le n} V_i(P_{i,k})\cdot \mathbf{x}_{i,k} &= \sum_{1\le k\le n} \Delta_k \cdot \mathbf{x}_{i,k} \\
&= \sum_{j\le k\le l} \Delta_k \cdot \mathbf{x}_{i,k} + \sum_{l'\le k\le n} \Delta_k \cdot \mathbf{x}_{i,k} \\
&\ge (\sum_{j\le k\le l} \Delta_k)\cdot \mathbf{x}_{i,l} + (\sum_{l'\le k\le n} \Delta_k)\cdot \mathbf{x}_{i,l'} \\
&= (\sum_{l'\le k\le n} \Delta_k)\cdot \mathbf{x}_{i,l'} - (\sum_{l'\le k\le n} \Delta_k)\cdot \mathbf{x}_{i,l} \quad \text{Inequality(8)} \\
&\ge 0,
\end{aligned}$$

Finally, let

$$\mathcal{L}(P_{i\downarrow}) = \{P_{i,j} \mid V_i(P_{i,j}) = V_i(P_{i,n})\}.$$

After each modification, either $V_i(P_{i,n})$ increases, or $|\mathcal{L}(P_{i\downarrow})|$ 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 $P'_{i\downarrow}$ after a finite number of modifications.

$\square$

Based on Lemma 7, in the rest of this paper we can assume that the optimal EMMS partitions are regular. In Lemma 8 we show that $\mathsf{LPT}_{i\downarrow}$ is also regular.

Lemma 8 *$LPT_{i\downarrow}$ is regular.*

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

□

In a regular partition $P_{i\downarrow}$, any bundle $P_{i,j}$ containing a huge item b has no other item. Otherwise, since $V_i(P_{i,j}) > V_i(b) \geq \mu$ and $\mu > V_i(P_{i,n})$, partition P is not regular. This fact about regular partitions (including $\mathsf{LPT}_{i\downarrow}$ and $O_{i\downarrow}$) allows us to deal with huge items. We are now ready to complete the proof Theorem 1.

Proof of Theorem 1. We use induction on the number of agents. For $n = 1$, the statement is trivial. For $n > 1$, we consider two cases. First, if $V_i(\mathsf{LPT}_{i,n}) \geq \mu/2$, by Lemma 6 we have

$$U_i(\mathsf{LPT}_{i\downarrow}, \omega) \geq \mathsf{EMMS}_i/2.$$

Therefore, we only need to consider the case that $V_i(\mathsf{LPT}_{i,n}) < \mu/2$. By Lemma 5, in such cases $\mathcal{M}$ includes at least one huge item.

Let $\mathcal{H}_O$ and $\mathcal{H}_{\mathsf{LPT}}$ be the set of the bundles containing huge items in $O_{i\downarrow}$ and $\mathsf{LPT}_{i\downarrow}$, respectively. Since both $O_{i\downarrow}$ and $\mathsf{LPT}_{i\downarrow}$ are regular, the bundles in $\mathcal{H}_O$ and $\mathcal{H}_{\mathsf{LPT}}$ do not contain anything but huge items, and each huge item is the only item within its bundle. Therefore, we have $\mathcal{H}_O = \mathcal{H}_{\mathsf{LPT}} = \mathcal{H}$. In addition, $\mathcal{H}$ are the $|\mathcal{H}|$ most valuable bundles in $\mathsf{LPT}_{i\downarrow}$ and therefore are allocated to the agents with the least influence on agent i. Otherwise, a very similar argument as in the proof of Lemma 5 yields $V_i(\mathsf{LPT}_{i,n}) \geq \mu/2$ which contradicts our assumption.

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

By the induction hypothesis, for this sub-instance, Inequality (1) 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,

$$U_i(\mathsf{LPT}_{i\downarrow}, \omega) \geq 1/2 \cdot U_i(O_{i\downarrow}, \omega') \geq 1/2 \cdot U_i(O_{i\downarrow}, \omega).$$

This, completes the proof of Theorem 1.

Notation	Description
$\mathcal{S}$	set of satisfied agents
ℓ_i	expectation level of agent i
M_i	A mapping from $\mathcal{S}$ to $O_{i\downarrow}$
$N_{i,j}$	Agents mapped to $O_{i,j}$ in M_i

Table 4 List of the notations mentioned in Section 6

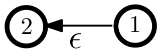


Fig. 7 The gap between MMS_i and EMMS_i may be large, even with very small externalities

Theorem 1 *For the network externalities model, we have*

$$U_i(\mathit{LPT}_{i\downarrow}, \omega) \geq \mathsf{EMMS}_i/2. \tag{6}$$

6 Approximate EMMS Allocation Problem

In this section, we focus on allocations that guarantee every agent i an approximation of EMMS_i. We start this section by comparing EMMS_i to MMS_i.

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

$$\mathsf{EMMS}_i = U_i(O_{i\downarrow}, \omega) \geq U_i(P_{i\downarrow}, \omega).$$

This, together with the fact that $U_i(P_{i\downarrow}, \omega) \geq \mathsf{MMS}_i$ (recall that we have $w_{i,i} = 1$) implies that $\mathsf{EMMS}_i \geq \mathsf{MMS}_i$ always holds. Now, in Lemma 9 we show that the gap between EMMS_i and MMS_i could be unbounded even for the instances with 2 agents.

Lemma 9 *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 7 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. For this instance, when $\epsilon < 1$ and $c \geq 1$ we have

$$\mathsf{EMMS}_1 = \min(c/\epsilon + \epsilon, 1 + c) = 1 + c$$

and $\mathsf{MMS}_1 = 1$. This means

$$\frac{\mathsf{EMMS}_1}{\mathsf{MMS}_1} \geq 1 + c > c.$$

□

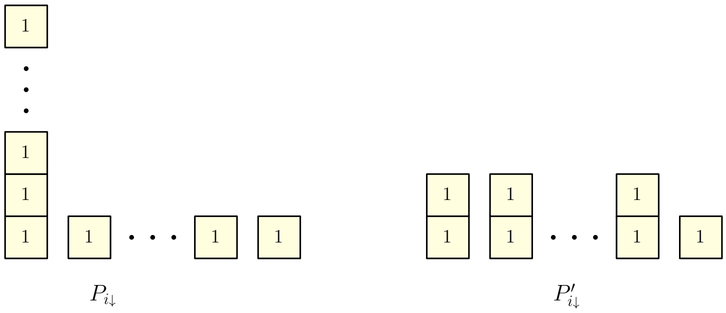


Fig. 8 For an instance with $2n-1$ items with value 1 to agnet i, $P_{i\downarrow}$ is a maximin partition. However, for influence vector $\mathbf{x}_{i\uparrow} = [1, 1, \ldots, 1, 0]$, utility gained by this partition is $n-1$ whereas the utility gained by the optimal EMMS partition ($P'_{i\downarrow}$) is $2n-3$.

The proof of Lemma 9 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. Note that in this example, the optimal partition for MMS value also happens to be optimal for EMMS value. But this is not always the case, see Figure 8 for an example.

Our main result is stated in Theorem 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 2 *Let $\mathbb{I}$ be an instance such that all the agents are α-self-reliant. Then, $\mathbb{I}$ admits an $\alpha/2$-*EMMS *allocation.*

In the rest of this section, we prove Theorem 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.

6.1 Bundle Claiming Algorithm (BC)

In this section, we present the ideas and a general description of the Bundle Claiming algorithm. We start by a simple observation. In Observation 2 we provide a simple bound which helps us prove our approximation guarantee.

Observation 2. *For every $k \leq n$, we have*

$$\sum_{k\leq j\leq n} \mathbf{x}_{i,j} \cdot \mathbf{v}_{i,j} \leq \mathbf{v}_{i,k} \cdot \sum_{1\leq j\leq n} \mathbf{x}_{i,j}.$$

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

$$\mathbf{v}_{i,4} \cdot \mathbf{x}_{i,4} + \mathbf{v} \cdot_{i,5} \mathbf{x}_{i,5} + \mathbf{v}_{i,6} \cdot \mathbf{x}_{i,6} \leq \mathbf{v}_{i,4} \cdot \sum_{1\leq j\leq n} \mathbf{x}_{i,j}.$$

Observation 2 is a direct result of the following two facts: first, for all $j > k$, we have $\mathbf{v}_{i,k} \geq \mathbf{v}_{i,j}$ and second,

$$\sum_{j>k} \mathbf{x}_{i,j} \leq \sum_{1\leq j\leq n} \mathbf{x}_{i,j}.$$

In our algorithm, we introduce a quantity ℓ_i for each agent i, which we refer to as the **expectation level** of agent i. We also refer to the quantity $\mathbf{v}_{i,\ell_i}/2$ as the **expectation value** of agent i. In the beginning of the algorithm, expectation level of each agent is set to 1 and therefore, the expectation value of each agent i is $\mathbf{v}_{i,1}$.

A simple flowchart of the algorithm is depicted in Figure 9. BC is consisted 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 $V_i(B) \geq \mathbf{v}_{i,\ell_i}/2$. We allocate B to one of the agents whose expectation is met (we say this agent is satisfied). Next, we update the expectation level of each remaining agent. The updating process is a fairly complex process which we precisely describe in Section 6.2. Roughly speaking, we increase 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 each remaining agent i, it is possible to partition the agents in $\mathcal{S}$ into ℓ_i subsets, namely $N_{i,1}, N_{i,2}, \ldots, N_{i,\ell_i-1}, N_{i,\ell_i}$, 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 $\mathbf{v}_{i,j}/2$ and at most $\mathbf{v}_{i,j}$ to agent i, and the total set of items allocated to the agents in N_{i,ℓ_i} is worth less than $\mathbf{v}_{i,\ell_i}/2$ 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 6.2, we perform the algorithm in a way that $\ell_i \leq n$ always holds for every agent i. We later show in Lemma 11 that if the external-satisfaction property holds for a remaining agent i with expectation level ℓ_i, total amount of externalities incurred by the satisfied agents on agent i is at least

$$\sum_{k<\ell_i} \mathbf{v}_{i,k} \cdot \mathbf{x}_{i,k}/2.$$

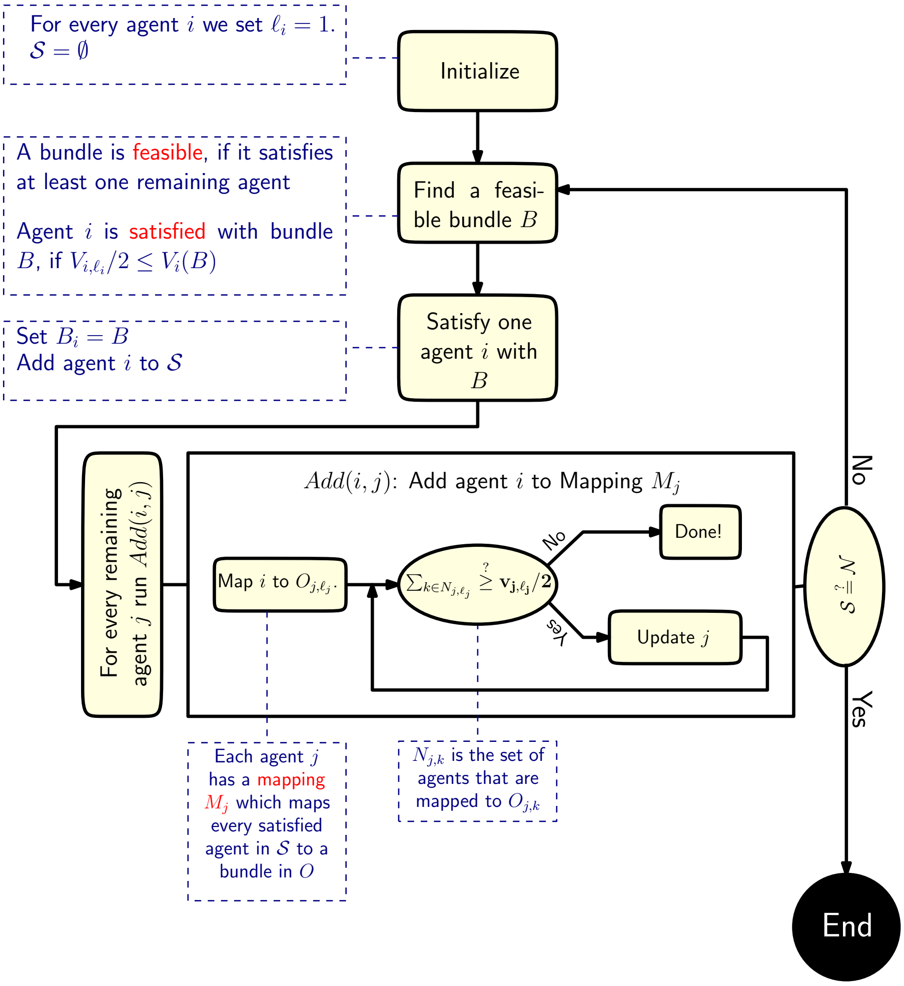


Fig. 9 A flowchart of BC Algorithm

The fact that EMMS_i is calculated with respect to the worst allocation of $O_{i\downarrow}$ is the key to prove this inequality.

Consider one step of the algorithm that a set B_i of items is allocated to agent i. Since B_i has met the expectation of agent i, $V_i(B_i) \geq \mathbf{v}_{i,\ell_i}/2$. Furthermore, as mentioned, the utility that agent i gained through the externalities of the satisfied agents is at least $\sum_{k<\ell_i} \mathbf{v}_{i,k} \cdot \mathbf{x}_{i,k}/2$. Assuming that agent i is α-self-reliant, his utility is at least

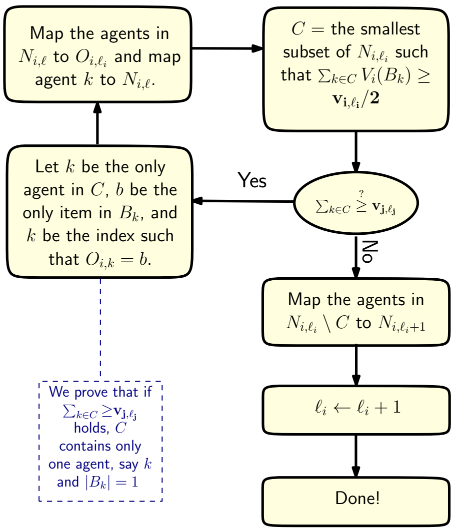


Fig. 10 A flowchart of the updating process

$$
\begin{aligned}
&\sum_{k<\ell_i} \mathbf{v}_{i,k} \cdot \mathbf{x}_{i,k}/2 + \mathbf{v}_{i,\ell_i}/2 && (w_{i,i} = 1)\\
\geq &\sum_{k<\ell_i} \mathbf{v}_{i,k} \cdot \mathbf{x}_{i,k}/2 + \Big(\sum_{k\geq\ell_i} \mathbf{v}_{i,k} \cdot \mathbf{x}_{i,k} / \sum_{1\leq j\leq n} \mathbf{x}_{i,j}\Big)/2 && \text{(Observation 2)}\\
\geq &(1/2 \sum_{1\leq j\leq n} \mathbf{x}_{i,j}) \cdot \sum_{k} \mathbf{v}_{i,k} \cdot \mathbf{x}_{i,k} && \Big(\sum_{1\leq j\leq n} \mathbf{x}_{i,j} \geq 1\Big)\\
\geq &(\alpha/2) \sum_{k} \mathbf{v}_{i,k} \cdot \mathbf{x}_{i,k}\\
= &(\alpha/2)\mathsf{EMMS}_i. && (9)
\end{aligned}
$$

Inequality (9) ensures that the total utility of agent i after allocating B_i is at least $(\alpha/2)\mathsf{EMMS}_i$. Therefore, our algorithm guarantees $\alpha/2\mathsf{EMMS}$ for the satisfied agents. Thus, it only remains to show that no agent remains unallocated at the end of the algorithm. We use the external-satisfaction property to prove this fact. We prove that in each step of the algorithm enough of items are remained to meet the expectation of each remaining agent. Consider agent i which has not satisfied yet. By external satisfaction property, value of the items allocated to the satisfied agents not in N_{i,ℓ_i} is

$$
\begin{aligned}
\sum_{k\in\mathcal{S}\setminus N_{i,\ell_i}} V_i(B_k) &= \sum_{1\leq j<\ell_i} \sum_{k\in N_{i,j}} V_i(B_k)\\
&\leq \sum_{1\leq j<\ell_i} \mathbf{v}_{i,j} && \text{(External satisfaction property)}
\end{aligned}
$$

ALGORITHM 1: Bundle Claiming algorithm

```
forall $a_j \in \mathcal{N}$ do
|   $\ell_j \leftarrow 1$                                  ▷ Initializing the expectation levels
end
while $\mathcal{S} \neq \mathcal{N}$ do
    forall $a_j \in \mathcal{N} \setminus \mathcal{S}$ do
    |   $B_j \leftarrow$ Minimum sized subset of $\mathcal{M}$, s.t. $V_j(B_j) \geq \mathbf{v}_{j,\ell_j}/2$;
    end
    $i \leftarrow \arg\min_i |B_i|$;
    Allocate $B_i$ to agent $i$ ;
    Add $i$ to $\mathcal{S}$ ;
    $\mathcal{M} \leftarrow \mathcal{M} \setminus B_i$ ;
    forall $j \in \mathcal{N} \setminus \mathcal{S}$ do
        $N_{j,\ell_j} \leftarrow N_{j,\ell_j} \cup \{i\}$. ;
        while $V_j(N_{j,\ell_j}) \geq 1/2 \cdot \mathbf{v}_{j,\ell_j}$ do
        |   Update(j)                                      ▷ (see Section 6.2)
        end
    end
end
```

Hence, the total value of the remaining items plus the items allocated to the agents in N_{i,ℓ_i} is at least

$$\sum_{1 \leq j \leq n} \mathbf{v}_{i,j} - \sum_{1 \leq j < \ell_i} \mathbf{v}_{i,j} = \sum_{\ell_i \leq j \leq n} \mathbf{v}_{i,j}.$$

Moreover, by the external satisfaction property, value of the items allocated to the agents in N_{i,ℓ_i} is less than $\mathbf{v}_{i,\ell_i}/2$. Thus, the value of the remaining items is at least

$$\sum_{\ell_i \leq j \leq n} \mathbf{v}_{i,j} - \mathbf{v}_{i,\ell_i}/2 \geq \mathbf{v}_{i,\ell_i}/2$$

which is enough to meet the expectation of agent i. Therefore, if we maintain the external satisfaction property for each remaining agent during the algorithm, all the agents will be satisfied by the end of the algorithm.

The detail of the bundle claiming algorithm is demonstrated in Algorithm 1. In the next section, we show how to maintain the external-satisfaction property.

Corollary 2 *Let $\mathbb{I}$ be an instance such that all the agents are α-self-reliant. Then, an $\alpha/4$-EMMS allocation for $\mathbb{I}$ can be found in polynomial time.*

Proof. Except the part related to finding the optimal EMMS partition, all other parts of BC algorithm can be trivially implemented in polynomial time. However, as mentioned earlier, finding the optimal EMMS partition in polynomial time is not possible unless $P = NP$.

To resolve this, we use $\mathsf{LPT}_{i\downarrow}$ instead of $O_{i\downarrow}$ in the updating process, while losing a factor 2 in the approximation guarantee. Note that the only property

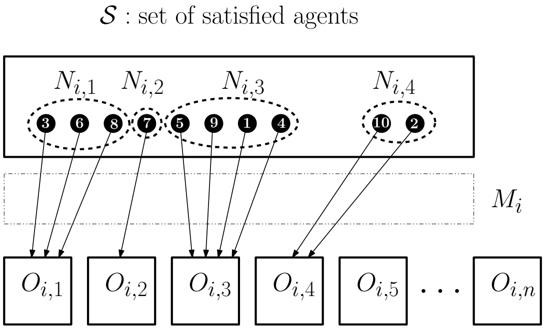


Fig. 11 Mapping M_i maps each satisfied agent to a bundle in $O_{i\downarrow}$.

of $O_{i\downarrow}$ that we use in the updating process is regularity. Recall that in Lemma 8 we proved $\mathsf{LPT}_{i\downarrow}$ is also regular. Therefore, using $\mathsf{LPT}_{i\downarrow}$ instead of $O_{i\downarrow}$ does not hurt the soundness of the algorithm. □

6.2 The External-satisfaction Property

In this section, we show how to maintain the external-satisfaction property. We start by giving a detailed explanation of the updating process. For each agent i in $\mathcal{S}$, we denote the bundle allocated to him by B_i. 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 a partitioning of $\mathcal{S}$ into ℓ_i bundles for agent i.

Definition 5 For every agent i, we define

$$M_i : \mathcal{S} \to \{O_{i,1}, O_{i,2}, \dots, O_{i,n}\}$$

as a mapping that corresponds each satisfied agent to a bundle of $O_{i\downarrow}$. Furthermore, we define $N_{i,j}$ as the set of agents that are mapped to $O_{i,j}$ in M_i. During the algorithm, we say mapping M_i is **valid**, if the following conditions hold:

(i) $\forall j < \ell_i \qquad \sum_{k \in N_{i,j}} V_i(B_k) \geq \mathbf{v}_{i,j}/2$
(ii) $\forall j < \ell_i \qquad \sum_{k \in N_{i,j}} V_i(B_k) \leq \mathbf{v}_{i,j}$
(iii) $\qquad \sum_{k \in N_{i,\ell_i}} V_i(B_k) < \mathbf{v}_{i,\ell_i}/2$
(iv) $\forall j > \ell_i \qquad$ No agent is mapped to $O_{i,j}$.

Figure 11 shows an example of M_i with $\ell_i = 4$. The validity conditions for this mapping is as follows:

$$\begin{aligned}
\mathbf{v}_{i,1}/2 \leq \quad & V_i(B_3) + V_i(B_6) + V_i(B_8) & \leq \mathbf{v}_{i,1} \\
\mathbf{v}_{i,2}/2 \leq \quad & V_i(B_2) & \leq \mathbf{v}_{i,2} \\
\mathbf{v}_{i,3}/2 \leq \quad & V_i(B_5) + V_i(B_9) + V_i(B_1) + V_i(B_4) & \leq \mathbf{v}_{i,3} \\
& V_i(B_{10}) + V_i(B_2) & < \mathbf{v}_{i,4}/2 \\
& N_{i,j} = \emptyset \text{ for all } j > 4 &
\end{aligned}$$

Note that each satisfied agent j is mapped to some bundle of $O_{i\downarrow}$ for every remaining agent i. Therefore, each satisfied agent is mapped to some bundle in $|\mathcal{N} \setminus \mathcal{S}|$ different mappings.

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 O_{j,ℓ_j} in M_j, i.e., we set $M_j(i) = O_{j,\ell_j}$. Throughout the algorithm, whenever the total value of the items allocated to agents in N_{j,ℓ_j} reaches $\mathbf{v}_{j,\ell_j}/2$ for some remaining agent j, M_j becomes invalid (condition (iii) is violated) 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 C of the agents in N_{j,ℓ_j} with the minimum size that satisfies conditions (i) and (ii) for N_{j,ℓ_j} and map the rest of the agents (i.e., agents in $N_{j,\ell_j} \setminus C$) to O_{j,ℓ_j+1}. Next, we increase ℓ_i by one.

Regarding the validity conditions of M_j, total value of the items allocated to the agents in C must be at least $\mathbf{v}_{j,\ell_j}/2$ and at most $\mathbf{v}_{j,\ell_j}$ (we call such subset a *compatible set*). If a compatible set C exists, we map the agents in $N_{j,\ell_j} \setminus C$ to O_{j,ℓ_j+1} in M_j and increase ℓ_j by one. However, there may be some cases that no subset of N_{j,ℓ_j} is compatible. For such cases, we use the argument in Lemma 10.

Lemma 10 *Suppose that total value of the items allocated to the agents in N_{j,ℓ_j} is at least $\mathbf{v}_{j,\ell_j}/2$, but N_{j,ℓ_j} 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,ℓ_j} contains at least one compatible subset.*

We defer the proof of Lemma 10 to Section 6.3. Using Lemma 10 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. Algorithm 2 shows a pseudo-code for this process. In addition, in Figure 10 you can find a flowchart of the updating process.

In the last part of this section, we prove Lemma 11 which shows that the value of the externalities imposed to agent i by the satisfied agents is lower-bounded by

$$\sum_{j<\ell_i} \mathbf{x}_{i,j} \cdot \mathbf{v}_{i,j}/2.$$

As said before, the fact that EMMS_i is defined with regard to the worst allocation of $O_{i\downarrow}$ plays a key role in proving Lemma 11.

Lemma 11 *Consider one step of the algorithm, and let agent i be an arbitrary remaining agent with $\ell_i > 1$. Then, we have*

$$\sum_{j \in \mathcal{S}} w_{i,j} \cdot V_i(B_j) \geq \sum_{1 \leq j < \ell_i} \mathbf{x}_{i,j} \cdot \mathbf{v}_{i,j}/2. \tag{10}$$

Proof. We show that in every step of the algorithm, for each remaining agent i, Inequality (10) holds. To prove this, we apply a sequence of exchanges between the bundles allocated to 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 (10) does not increase. [5] Next, we show that after these exchanges, Inequality (10) still 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 $w_{i,j}$). First, we allocate the items 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 (10).

In addition, let agent j' be the agent with $w_{i,j'} = \mathbf{x}_{i,1}$. Since agent j' has the minimum influence on i among all the agents, we have $w_{i,j'} \leq w_{i,j}$. Now, let B_j and $B_{j'}$ be the current bundles of agents j and j' ($B_{j'}$ might be empty). 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 (10) since we have $w_{i,j'} \leq w_{i,j}$. 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

$$\sum_{j \in S} w_{i,j} \cdot V_i(B_j).$$

Furthermore, one can easily verify that despite a possible decrement in the total value of the items allocated to the agents in $N_{i,r}$, since before this exchange we had $V_i(B_j) \geq \mathbf{v}_{i,1}/2 \geq \mathbf{v}_{i,r}/2$, after these exchanges condition (i) still holds for $N_{i,r}$. In Figure 12 you can find an example of this process.

We repeat the same procedure for $N_{i,2}, N_{i,3}, \ldots, N_{i,\ell_i-1}$. After this sequence of exchanges, each $N_{i,j}$ contains one agent j', where $w_{i,j'} = \mathbf{x}_{i,j}$. Furthermore, after these exchanges, the first condition for the validity of M_i holds and hence, the value of the items in the bundle of agent j' for agent i is at least $\mathbf{v}_{i,j}/2$. Therefore, total amount of the externalities of the satisfied agents is at least

$$\sum_{1 \leq j < \ell_i} \mathbf{x}_{i,j} \cdot \mathbf{v}_{i,j}/2.$$

□

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

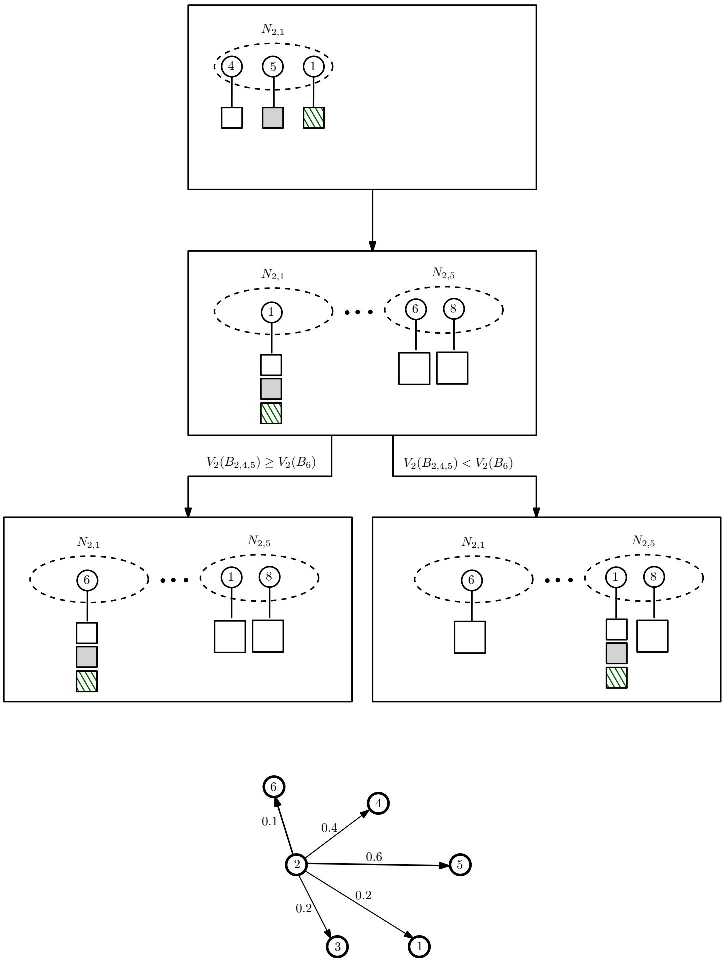


Fig. 12 An example of the process introduced in Lemma 11. Suppose that the influences on agent 2 is as illustrated in the graph. To perform the process described in Lemma 11 on agent 2, we proceed as follows: first we allocate bundles B_4 and B_5 to agent 1, which has the least influence among agents $\{1,4,5\}$ on agent 2 (we denote the new bundle by $B_{1,4,5}$). Next, we find the agent whose influence on agent 2 equals $\mathbf{x}_{2,1} = 0.1$, which is agent 6. We exchange the set that agent 1 and 6 belong to, so that agent 6 is moved to $N_{2,1}$ and agent 1 is moved to $N_{2,5}$. Next, we compare values of $B_{1,4,5}$ and B_6. Between these two bundles, the one with the less value for agent 2 is allocated to agent 6 and the other one is allocated to agent 1.

6.3 Proof of Lemma 10.

In this section, we prove Lemma 10, which is the most challenging technical part of the paper. Before going through the proof, we recall the statement of this lemma.

Lemma 10 *Suppose that total value of the items allocated to the agents in N_{j,ℓ_j} is at least $\mathbf{v}_{j,\ell_j}/2$, but N_{j,ℓ_j} 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,ℓ_j} contains at least one compatible subset.*

In the rest of this section, we prove Lemma 10. Let C be a subset of N_{j,ℓ_j} with the minimum size that satisfies condition (i). Such a set trivially exists. However, since no subset of N_{j,ℓ_j} is compatible, we have

$$\sum_{k \in C} V_j(B_k) > \mathbf{v}_{j,\ell_j}.$$

Since C is minimal, no proper subset of C satisfies condition (i). It is easy to observe that this can only happen when C contains only one agent, say k, with $V_j(B_k) > \mathbf{v}_{j,\ell_j}$. We later show in Lemma 12 that $|B_k| = 1$; but for now suppose that $|B_k| = 1$ holds and b is the only item in B_k.

Since $O_{j\downarrow}$ is regular and $V_j(b) > \mathbf{v}_{j,\ell_j}$, 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 N_{j,ℓ_j}. After this exchange, for $N_{j,\ell}$ we have

$$\sum_{i \in N_{j,\ell}} V_j(B_i) = V_j(O_{j,\ell}) = \mathbf{v}_{j,\ell}$$

and therefore, conditions (i), (ii) preserve for $N_{j,\ell}$ after this process.

Again, if no subset of N_{j,ℓ_j} is compatible, we repeat this modification. Each time we modify M_j, the number of indices ℓ for which $N_{j,\ell}$ contains an agent k with $O_{j,\ell} = B_k$ increases by one. Therefore, the process terminates after a finite number of modifications. □

Algorithm 2 illustrates an overview of the updating procedure, which completes the BC algorithm. It only remains to show the fact that $|B_k| = 1$, which we prove in Lemma 12.

ALGORITHM 2: Update M_j

Resolve $= 0$
while *Resolve* $== 0$ **do**
  $C \leftarrow$ Minimum sized subset of N_{j,ℓ_j}, s.t. $\sum_{k \in C} V_j(B_k) \geq 1/2 \cdot \mathbf{v}_{j,\ell_j}$
  if $\sum_{k \in \delta} V_j(B_k) \leq \mathbf{v}_{j,\ell_j}$ **then**
    foreach $k \in N_{j,\ell} \setminus C$ **do**
      $M_j(k) = O_{j,\ell+1}$
    end
    $\ell_j \leftarrow \ell_j + 1$
    Resolve $+= 1$ ▷ Resolved
  else
    Let ℓ be an index s.t. $O_{j,\ell} = B_k$, where $C = \{k\}$.
    Swap C (which is a subset of N_{j,ℓ_j}) with $N_{j,\ell}$. ▷ One step closer to resolve
  end
end

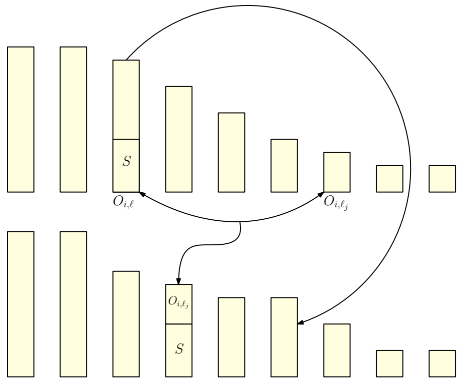


Fig. 13 If $|O_{j,\ell}| > 1$ and $\mathbf{v}_{j,\ell_j} < \mathbf{v}_{j,\ell}/2$ simultaneously hold, we can improve the partition by adding a subset $S \in O_{j,\ell}$ to O_{j,ℓ_j}.

Lemma 12 *Suppose that the total value of the items allocated to the agents in N_{j,ℓ_j} is at least $\mathbf{v}_{j,\ell_j}/2$, but N_{j,ℓ_j} admits no compatible subset, and let C be the subset of N_{j,ℓ_j} with minimum size that satisfies condition (i). Then, C contains only one agent, say agent k where $|B_k| = 1$.*

Proof. As mentioned in Lemma 10, it is easy to observe that $|C| = 1$. Here, we argue that if agent k is the only agent in C, then $|B_k| = 1$. As a contradiction, let z be the first step of the algorithm that the condition of Lemma 12 is violated, i.e.,

- Total value of the items allocated to the agents in N_{j,ℓ_j} is at least $\mathbf{v}_{j,\ell_j}/2$.
- N_{j,ℓ_j} admits no compatible subset.
- C (the smallest subset of N_{j,ℓ_j} satisfying condition (i)) contains only one agent say k.
- $|B_k| > 1$.

In addition, let z' 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'. Trivially, we have $z' \leq z$ and $\ell'_j \leq \ell_j$.

Notation	Description	Expectation level
z	First step that $C = \{k\}$, but $\|B_k\| > 1$	ℓ_j
$z' \leq z$	The step that B_k is allocated to k	$\ell'_j \leq \ell_j$

Table 5 Two steps z and z'.

Lemma 13 *We can suppose w.l.o.g that either $\mathbf{v}_{j,\ell_j} \geq \mathbf{v}_{j,\ell'_j}/2$ or we have $|O_{j,\ell}| = 1$ for all $\ell \leq \ell'_j$.*

Proof. 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 $V_j(S) \leq \mathbf{v}_{j,\ell}/2$. Now, if

$$\mathbf{v}_{j,\ell_j} < \mathbf{v}_{j,\ell'_j}/2 < \mathbf{v}_{j,\ell}/2$$

holds, by a very same reasoning as in Lemma 7, adding S to bundle O_{j,ℓ_j} yields a new partition which is at least as good as $O_{j\downarrow}$ (see Figure 13). $\square$

Regarding Lemma 13, we consider two cases.

First, assume that $|O_{j,\ell}| = 1$ for all $\ell \leq \ell'_j$. For this case, since the expectation level of agent j in step z' is ℓ'_j, regarding conditions (ii) and (iii) we have

$$\sum_{i \in \mathcal{S}} V(B_i) < \sum_{1 \leq \ell < \ell'_i} \mathbf{v}_{j,\ell} + \mathbf{v}_{j,\ell'_j}/2 < \sum_{1 \leq \ell \leq \ell'_i} \mathbf{v}_{j,\ell}$$

Therefore, at least one of the items in $\bigcup_{\ell \leq \ell'_j} O_{j,\ell}$ is not allocated to any satisfied agent before step z', and this item alone meets the expectation of agent j. This contradicts the fact that at step z', B_k was the set with the minimum size that meets the expectation of a remaining agent (note that we supposed $|B_k| > 1$).

Second, assume that $\mathbf{v}_{j,\ell_j} \geq \mathbf{v}_{j,\ell'_j}/2$. In step z', the expectation value of agent j equals $\mathbf{v}_{j,\ell'_j}/2$. Furthermore, $V_j(B_k) > \mathbf{v}_{j,\ell_j}$ which means $V_j(B_k) > \mathbf{v}_{j,\ell'_j}/2$ and hence, the second condition holds for B_k in step z'. On the other hand, $V_j(B_k) < \mathbf{v}_{j,\ell'_j}$, otherwise since $|B_k| > 1$ a proper subset of B_k would meet the expectation of agent j in step z'. Therefore, in step z', $C = \{k\}$ is the only compatible set for updating M_j and hence, agent k is mapped to O_{j,ℓ'_j} in step z'. This also implies that $z' \neq z$, since we supposed that no compatible subset exists in step z.

Furthermore, notice that since $|B_k| > 1$, no item could alone meet the expectation of any agent, including agent j in step z' (recall that in each step, we select a bundle with the minimum size that meets the expectation of a remaining agent). This means that every remaining item in step z' has the value less than $\mathbf{v}_{j,\ell'_j}/2$. Therefore, value of every remaining item after step z' for agent j is less than $\mathbf{v}_{j,\ell}/2$ for every $\ell \leq \ell'_j$.

On the other hand, in all the modifications (based on Lemma 10) after step z' and before step z, the bundle of the agent in the selected compatible set C is consisted of only one item (z is te first step that the size of the bundle allocated to the agent in C is more than 1). According to the way we modify M_j (based on Lemma 10), after step z', no modification affects the agents that are mapped to bundles $O_{j,\ell}$ for $\ell \leq \ell'_j$. Because each single item bundle after step z' has a value less than $\mathbf{v}_{j,\ell}/2$ for every $\ell \leq \ell'_j$. But this contradicts the fact that agent k is mapped to N_{j,ℓ_j} in step z, because agent k was mapped to O_{j,ℓ'_j} in step z' and no modification changes $M_j(k)$.

$\square$

7 Discussion and Future Directions

An exciting open direction is to find approximation allocation algorithms for the general externalities model, with no restriction on the value of $V_{i,j}(b_k)$. 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 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 α, such that there exists a partition P of the items in which $\omega_i(P)$ is a α-approximation of $\omega_i(O_{i\downarrow})$ for every agent i. Note that if such a partition exists, any allocation of it to the agents is a α-EMMS allocation.

Another open question is to present a $PTAS$ for finding the optimal EMMS partition for an agent in the network externalities model.

References

1. Georgios Amanatidis, Georgios Birmpas, George Christodoulou, and Evangelos Markakis. Truthful allocation mechanisms without payments: Characterization and implications on fairness. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 545–562. ACM, 2017.
2. Georgios Amanatidis, Georgios Birmpas, and Evangelos Markakis. On truthful mechanisms for maximin share allocations. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 31–37. AAAI Press, 2016.
3. 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.
4. 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.
5. Haris Aziz, Gerhard Rauchecker, Guido Schryen, and Toby Walsh. Algorithms for max-min share fair allocation of indivisible chores. In *AAAI Conference on Artificial Intelligence*, volume 17, pages 335–341. AAAI Press, 2017.
6. Siddharth Barman, Arpita Biswas, Sanath Kumar Krishnamurthy, and Yadati Narahari. Groupwise maximin fair allocation of indivisible goods. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 917–924. AAAI Press, 2018.
7. Siddharth Barman and Sanath Kumar Krishna Murthy. Approximation algorithms for maximin fair division. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 647–664. ACM, 2017.
8. 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.
9. Simina Brânzei, Ariel D Procaccia, and Jie Zhang. Externalities in cake cutting. In *International Joint Conference on Artificial Intelligence*, pages 55–61, 2013.

10. Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.
11. Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.
12. Bryan L Deuermeyer, 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.
13. 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.
14. Mohammad Ghodsi, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods: Improvements and generalizations. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 539–556. ACM, 2018.
15. Laurent Gourvès and Jérôme Monnot. Approximate maximin share allocations in matroids. In *International Conference on Algorithms and Complexity*, pages 310–321. Springer, 2017.
16. Ronald L. Graham. Bounds on multiprocessing timing anomalies. *SIAM journal on Applied Mathematics*, 17(2):416–429, 1969.
17. 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.
18. Godfrey Harold Hardy, John Edensor Littlewood, György Pólya, DE Littlewood, G Pólya, et al. *Inequalities*. Cambridge university press, 1952.
19. Tobias Heinen, Nhan-Tam Nguyen, Trung Thanh Nguyen, and Jörg Rothe. 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*, pages 1–38, 2018.
20. Miho Hong and Jaeok Park. Core and top trading cycles in a market with indivisible goods and externalities. 2018.
21. David Kempe and Mohammad Mahdian. A cascade model for externalities in sponsored search. In *International Workshop on Internet and Network Economics*, pages 585–596. Springer, 2008.
22. 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.
23. David Kurokawa, Ariel D Procaccia, and Junxing Wang. When can the maximin share guarantee be guaranteed? In *AAAI Conference on Artificial Intelligence*, volume 16, pages 523–529. AAAI Press, 2016.
24. 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.
25. Minming Li, Jialin Zhang, and Qiang Zhang. Truthful cake cutting mechanisms with externalities: do not make them care for others too much! In *International Joint Conference on Artificial Intelligence*, pages 589–595, 2015.
26. 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.
27. 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.
28. Hugo Steinhaus. The problem of fair division. *Econometrica*, 16(1), 1948.
29. Warut Suksompong. Approximate maximin shares for groups of agents. *Mathematical Social Sciences*, 92:40–47, 2018.
30. Rodrigo A Velez. Fairness and externalities. *Theoretical Economics*, 11(1):381–410, 2016.
31. Gerhard J Woeginger. A polynomial-time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters*, 20(4):149–154, 1997.