Fair and Efficient Resource Allocation with Externalities

Shaily Mishra  
International Institute of Information Technology (IIIT)  
Hyderabad, India  
shaily.mishra@research.iiit.ac.in

Manisha Padala  
International Institute of Information Technology (IIIT)  
Hyderabad, India  
manisha.padala@research.iiit.ac.in

Sujit Gujar  
International Institute of Information Technology (IIIT)  
Hyderabad, India  
sujit.gujar@iiit.ac.in

ABSTRACT
In resource allocation, it is common to assume that agents have a utility for their allocated items and zero utility for unallocated ones. We refer to such valuation domain as 1-D. This assumption of zero utility for unallocated items is not always valid. For example, in the pandemic, allocation of ventilators, oxygen beds, and critical medical help yields dis-utility to an agent when not received in time, i.e., a setting where people consume resources at the cost of others’ utility. Various externalities affect an agent’s utility, i.e., when an agent doesn’t receive an item, it can result in their gain (positive externalities) or loss (negative externalities). The existing preference models lack capturing the setting with these externalities. We conduct a study on a 2-D domain, where each agent has a utility (\( u \)) for an item assigned to it and utility (\( u' \)) for an item not allocated to it. We consider a generalized model, i.e., goods and chores. There is vast literature for allocating resources both fairly and efficiently. We observe that adapting the existing notions of fairness and efficiency to the 2-D domain is non-trivial.

We propose a utility transformation (\( T_u \)) and valuation transformation (\( T_v \)) to convert from the 2-D domain to 1-D, i.e., the existing domain. We study the retention of fairness and efficiency property given this transformation, i.e., an allocation with property \( P \) in a 1-D domain also satisfies property \( P \) in 2-D, and vice versa. If a property is retainable, we can apply the transformation, and all the existing approaches are valid for the 2-D domain. Further, we study whether we can apply current results in a 2-D domain when an agent doesn’t receive an item, it can result in their gain (positive externalities) or loss (negative externalities). The existing approaches are valid for the 2-D domain. Further, we study fairness and efficiency notions such as Pareto Optimality, Utilitarian Welfare, Nash Welfare, and Egalitarian Welfare.

KEYWORDS
Fairness, Efficiency, Externalities, Resource Allocation

1 INTRODUCTION
Division of resources among interested agents is well-explored by researchers. Economists have proposed various fairness and efficiency notions widely applicable in real-world settings, such as division of inheritance, land, task, vaccines, etc. [12, 33, 37–39]. Computer Scientists, along with these notions, also explored computational aspects of some widely accepted fairness notions [4, 5, 9, 16, 17, 20, 35]. Such endeavors have led to web-based applications that offer readily available solutions, such as Spliddit \(^1\), The Fair Proposals System \(^2\), Courism \(^3\), Divide Your Rent Fairly \(^4\), etc. In these applications, all the agents provide their valuations towards each item as input. The underlying algorithm aggregates the valuations and provides allocations that are fair and/or efficient as required. The algorithms used focus on resource allocation without externalities, i.e., an agent’s utility for not receiving a resource is zero. We believe this is inadequate when modeling resource allocation for necessary commodities.

The resources can be goods, chores, or a combination of both. Agents would want goods, i.e., they obtain a positive valuation for good for the resource while avoiding chores (negative valuation). Some resources could be goods for one and chore for another, which we call combination. Consider the situation during the Covid pandemic; there is a sudden and steep requirement for life-supporting resources like hospital beds, ventilators, and vaccines. Getting a free covid vaccination affects an agent positively. Even if someone else gets the free vaccination instead of the agent, the agent values it positively, possibly less. The more vaccinated people, the better the situation. If an agent does not receive resources like a ventilator bed can incur a loss. Thus, the valuation of an agent may not only depend on obtaining the resource but also on not getting the resource. Such an effect of external factors on agents’ utility is captured via externalities.

Externality in valuation signifies the effect of external factors on agents’ utility. The agent may incur a benefit—positive externalities or cost—negative externalities for an unassigned resource. In the most general form of externalities, the utility of not receiving an item depends on which other agent receives it. With \( n \) agents, each agent’s valuations are \( n \)-dimensional; hence we refer to this as a \( N \)-D domain. In the absence of externalities, it is a 1-D domain. In a \( N \)-D domain, the agent’s utility becomes complex, and in [13], the authors focus only on positive externalities. In this work, which we motivate towards the pandemic, considering both positive and negative externalities is important. We consider externalities such that the agents incur a cost/benefit for not receiving a resource,
yet it is independent of which other agent receives it. We refer to such a valuation domain as 2-D domain – valuation \( v \) for a bundle of items if an agent receives it, and valuation \( v' \) otherwise. If \( v' \) is positive, it is a positive externality, and if \( v' \) is negative, it is a negative externality. In 1-D domain, \( v' \) is zero. In this work, we focus on fair and efficient resource allocation in a 2-D domain for indivisible goods or/and chores. There are widely studied fairness and efficiency notions in a 1-D domain, which we adapt to 2-D.

**Fairness notions.** Fair allocation of goods and chores is well studied for 1-D domain \([2, 4, 5, 16, 21, 22]\). The most desirable notions of fairness we consider are Envy-freeness (EF), which ensures that no agent has higher utility for a bundle received by another agent \([19]\). Equitability (EQ) ensures that agents have equal utilities for their shares \([18]\), Proportionality (PROP) ensures that every agent receives at least \(1/n\) of its valuation of the entire bundle \([38]\), and Maximin Share (MMS) ensures that each agent receive at least its MMS value \([14]\).

We cannot ignore externalities in a 2-D domain if we want to ensure the above properties. For example, consider two agents \([1, 2]\), and two goods \(\{g_1, g_2\}\). Agent 1 has valuations, represented as \((v, v')\), \(g_1: (6, -1), g_2: (5, -100)\) and Agent 2 has \(g_1: (5, -50), g_2: (6, -1)\). If we consider 1-D valuation, allocating \(g_1\) to 1 and \(g_2\) to 2 is EF, which is not EF in a 2-D domain because agent 1 envies agent 2. Similarly, it is crucial to note that the existing definition of PROP does not capture the essence of proportional allocation in the 2-D domain; for example, it does not consider the dis-utility of not receiving good in case of a negative externality. For the above example, if we only consider the proportionality definition, each agent should receive goods worth at least \(11/2\). Guaranteeing this amount is impossible in a 2-D domain. We define PROP-E, an extension of proportionality for resource allocation with externalities. PROP-E reduces to the Average-share definition proposed in \([36]\) for additive valuations, and both these definition reduces to proportionality under a 1-D domain (Section 2). We conduct a study for the possibility and challenges of ensuring these fairness notions and their approximations for a 2-D domain.

**Efficiency notions.** Efficiency is another essential aspect of resource allocation. Common efficiency notions like Pareto Optimality (PO), Maximum Utilitarian Welfare (MUW), Maximum Nash Welfare (MNW), and Maximum Egalitarian Welfare (MEW) also require further analysis in a 2-D domain.

### 1.1 Our Contributions

In this paper, we study the 2-D domain for indivisible goods or chores with positive and negative externalities for different fairness (EF, PROP-E, EQ, and MMS) and efficiency notions (PO MUW, MNW, and MEW).

We aim to adapt the algorithms designed for obtaining the above properties in a 1-D domain. Towards this, our first approach proposes a reduction from a 2-D domain to 1-D through certain transformations such that ensuring fairness or efficiency in the transformed valuations also ensures them in the original 2-D domain. In 2-D domain, for general valuations we propose \(T_{\text{eq}}\) (Definition 9), and for additive valuations, we propose \(T_{\text{eq}}\) (Definition 10). Our second approach is to verify if an algorithm proposed for 1-D domain can be directly used without transformation. In particular,

- EF and its relaxations like EF1, EFX can be retained through the transformation; hence we can adapt all the existing algorithms with appropriate transformation (Proposition 2).
- We define PROP-E (Definition 4), an extension of PROP to suit the 2-D domain. We show that PROP-E reduces to Average-share as defined in \([36]\) (Proposition 4). Finally, we show that PROP-E is also retained through transformation for additive valuations (Proposition 5).
- We observe that we cannot retain EQ and its relaxations. We show that the algorithm proposed in \([26]\) for 1-D can be directly used in 2-D (Proposition 3) without requiring any transformation.
- Lastly, we study MMS and show that while we can retain MMS using transformation, we cannot retain \(a\)-MMS under a specific setting. The algorithm proposed in \([7, 23–25, 27]\) cannot be applied directly in 2-D, leaving scope to explore more in this domain. (Propositions \([6, 7, 8, 9, 10]\))
- Efficiency Notions: We show that we can retain Pareto Optimality (PO) and Maximal Utilitarian Welfare (MUW) using transformation (Propositions \([11, 12]\)). We cannot define Maximum Nash Welfare (MNW), and we cannot retain Maximum Egalitarian Welfare (MEW) in a 2-D domain. (Proposition 13)

### 1.2 Related Work

There are few papers that consider externalities in resource allocation. Bramie, Procaccia, and Zhang \([13]\) and Seddighin, Saleh, and Ghodsi \([36]\) study allocating divisible and indivisible goods with positive externalities. Authors in \([13]\) generalize notions of proportionality and envy-freeness indivisible goods allocation with positive externalities. Authors in \([36]\) explore MMS allocation for goods allocation with positive externalities; it generalizes maximin share as EMMS and further provides approximation algorithms for calculating EMMS and EMMS allocations. \([31]\) analyze facility location games with externalities. To the best of our knowledge, there is no relevant work that considers negative externalities.

We summarize the existing algorithms for 1-D valuations available for each of the fairness notions.

Envy-freeness. EF may not exist for indivisible items, for example, two agents and one good. Hence we consider two popular relaxations of EF, Envy-freeness up to one item (EF1) \([14, 32]\) and Envy-freeness up to any item(EFX) \([16]\). We have poly-time algorithms to find EF1 in general monotone valuations for goods \([32]\) and chores \([11]\). For additive valuations, EF1 can be found using Round Robin \([16]\) in goods or chores, and Double Round Robin \([3]\) in combination. When the valuations are general, non-monotone, and non-identical, a poly-time algorithm is given by \([15]\) for two agents. Authors in \([34]\) present an algorithm to find EFX allocation under identical general valuations for goods.

Equitability. Relaxation to EQ are EQ1 and EQX \([14, 16]\). Authors in \([26]\) present a polynomial-time algorithm to find EQX allocation. They present a pseudo-polynomial time algorithm to find EQ1 and PO allocation in goods and chores in \([20, 22]\).

Proportionality. We have PROP1 and PROPX as the relaxation to PROP. For additive valuations, EF1 implies PROP1, and EFX implies PROPX.
MMS. MMS allocations do not always exist [29, 35]. The papers [1, 8, 23, 35] showed that 2/3-MMS for goods always exists. [24, 25] showed that 3/4-MMS for goods always exists. Authors in [7] presented a polynomial-time algorithm for 2-MMS for chores. The algorithm presented in [8] gives 4/3-MMS for chores. Authors in [27] showed that 11/9-MMS for chores always exists. Authors in [28] explored α-MMS for a combination of goods and chores.

**Fair and Efficient.** In [16], the authors showed that MNW allocation is EF1 and PO for indivisible goods and gave a pseudo-polynomial time algorithm [9]. For a combination of resources, the authors in [3] presented a polynomial-time algorithm to find EF1 and PO for two agents. An Algorithm to find PROP1 and PO was proposed by [6] for a combination of resources. Authors in [5] proposed a pseudo-polynomial time for finding Utilitarian maximizing among EF1 or PROP1 in goods.

**Organization.** In Section 2, we define all the notions and definitions related to fairness and efficiency used in this paper. In Section 4, we present results related to preserving fairness under transformation along with algorithmic analysis. In Section 5, we study efficiency preservation upon transformation and its corresponding algorithmic analysis.

## 2 PRELIMINARIES

We consider a resource allocation problem \((N, M, V)\) for determining an allocation of \(M = [m]\) indivisible items among \(N = [n]\) interested agents, \(m, n \in \mathbb{N}\).

- An allocation \(A \in [0, 1]^{n \times m}\) is an n-way partition \((A_1, \ldots, A_n)\) of \(M\). Here, \(A_i \subseteq [m]\) is the bundle assigned to agent \(i\) and \(\forall i, j \in N\) and \(i \neq j, A_i \cap A_j = \emptyset\). We consider a complete allocation of items, i.e., \(\bigcup_i A_i = M\). We represent \(A_i\) as \(\bigcup_{j \in i \in N} A_j\).
- Typically in a 1-D domain, agents have valuation function \(v_i : 2^M \rightarrow \mathbb{R}\) for items. In 2-D domain, \(V = \{V_1, V_2, \ldots, V_n\}, \forall i \in N\), and for subset \(S \subseteq M, v_i(S) = (v_i(S), v'_i(S))\). For goods, \(v_i : 2^M \rightarrow \mathbb{R}_{\geq 0}\), and for chores, \(v_i : 2^M \rightarrow \mathbb{R}_{\leq 0}\). For positive externality, \(v_i^p : 2^M \rightarrow \mathbb{R}_{\geq 0}\), and for negative externality, \(v_i^n : 2^M \rightarrow \mathbb{R}_{\leq 0}\).
- The valuation naturally induces utility structure to the agents, i.e., the utility of an agent \(i\) for a bundle \(S\) in a 2-D domain \(V\) as, \(u_i^{V}(S) = u_i(S) + v_i'(M \setminus S)\). Also, \(u_i^{V}(\emptyset) = 0 + v_i'(M)\). In the case of 1-D domain and \(\forall S \subseteq M, v_i(S) = v_i(S)\) and \(v_i'(S) = 0\), and utility \(u_i(S) = v_i(S)\).
- For additive valuations, \(u_i^{V}(S) = \sum_{k \in S} v_i((k)) + \sum_{k \in S} v_i'(\{k\})\). Note that, in 2-D, additive valuations does not imply additive utility, i.e., \(u_i^{V}(A_1) = u_i(A_1) + v_i(M \setminus A_1)\), and \(u_i^{V}(A_2) = u_i(A_2) + v_i(M \setminus A_2)\). However, \(u_i^{V}(A_1 \cup A_2) \neq u_i^{V}(A_1) + u_i^{V}(A_2)\). Also when agents have identical valuations, \(\forall j, n \in N, S \subseteq M, u_i(S) = u_j(S)\).
- We assume monotonicity for goods, i.e., \(\forall S \subseteq T \subseteq M, u_i^{V}(S) \leq u_i^{V}(T)\). Similarly, we assume anti-monotonicity for chores, i.e., \(\forall S \subseteq T \subseteq M, u_i^{V}(S) \geq u_i^{V}(T)\). As we increase goods in an agent’s bundle, the utility increases, and as we increase chores, the utility decreases. Note that we use the notation \(v_{ik}\) or \(v_i((k))\) to represent the valuation of agent \(i\) for item \(k\).

We now formally define the fairness and efficiency criteria that we analyze for the 2-D domain.

### Definition 1 (Envy-free (EF) and relaxations [3, 14–16, 19]).

An allocation \(A\) that satisfies,
\[\forall i, j \in N, u_i(A_j) = u_i(A_j) \text{ is } EF\] (1)
\[\forall k \in A_i \text{ s.t. } v_{ik} < 0, u_i(A_j \setminus \{k\}) \geq u_i(A_j)\] is EFX (2)
\[\exists k \in \{A_i \cup A_j\} \text{ s.t. } u_i(A_j \setminus \{k\}) \geq u_i(A_j) \text{ is EF}\] (3)

Informally, A is EF when agent envies no other. It is EF1, i.e., Envy-free up to one item, if each agent’s envy can be eliminated by either removing a good from the envied agent’s allocation or removing a chore from its allocation. For EFX, i.e., Envy-free up to any item, the envy is removed when any good or chore is removed. Hence EF \(\implies\) EFX \(\implies\) EF1.

### Definition 2 (Equitable (EQ) and relaxations [18, 21, 22]).

An allocation \(A\) is said to be equitable, when \(\forall i, j \in N, u_i(A_j) = u_i(A_j)\). An allocation \(A\) is said to be EQ1, i.e., Equitable up to one item, \(u_i(A_j \setminus \{k\}) \geq u_i(A_j \setminus \{k\})\), \(\exists k \in \{A_i \cup A_j\}\). An allocation \(A\) is said to be EQX, i.e., Equitable up to any item, \(u_i(A_j) \geq u_i(A_j \setminus \{k\})\), \(\forall k \in A_j\) and \(v_{ik} \geq 0\), and \(u_i(A_j \setminus \{k\}) \geq u_i(A_j)\), \(\forall k \in A_j\), and \(v_{ik} \leq 0\).

Informally, in EQ, all agents are treated equally, i.e., everyone values their bundles equally. It is EQ1, i.e., Equitable up to one good, if the utility of an agent \(i\) is less than other agents. There exists some good in other agents’ bundle or some chore in its bundle; on removing it, the utility of agent \(i\) is at least the utility of other agents. Similarly, for EQX, the above satisfies all goods in other’s bundles and for all chores in their bundles.

### Definition 3 (Proportionality (PROP) [38]).

An allocation \(A\) is said to be proportional, if \(\forall i \in N, u_i(A_i) \geq 1/n \cdot u_i(M)\).

For 2-D domain, achieving PROP is impossible (e.g., given in Section 1). Hence, we adapt the definition and propose PROP-E for valuations with externalities as follows.

### Definition 4 (Proportionality with externality (PROP-E)).

An allocation \(A\) is said to PROP-E if
\[\forall i \in N, u_i(A_i) \geq 1/n \cdot \sum_{j \in N} u_j(A_j)\] (4)

We now define the relaxations for PROP-E as below, similar to the relaxations of PROP.

### Definition 5 (PROP-E relaxations).

An allocation \(A\) that satisfies,
\[\forall k \in \{M \setminus A_i\}, v_{ik} > 0, u_i(A_i \cup \{k\}) \geq 1/n \sum_{j \in N} u_j(A_j)\] or
\[\forall k \in A_i, v_{ik} < 0, u_i(A_i \setminus \{k\}) \geq 1/n \sum_{j \in N} u_j(A_j)\] is PROPX-E (5)
\[\exists k \in \{M \setminus A_i\}, u_i(A_i \cup \{k\}) \geq 1/n \sum_{j \in N} u_j(A_j),\] or
\[\exists k \in A_i, u_i(A_i \setminus \{k\}) \geq 1/n \sum_{j \in N} u_j(A_j)\] is PROPI-E (6)

### Definition 6 (Maxmin Share MMS [14]).

An allocation \(A\) is said to be MMS if \(\forall i \in N, u_i(A_i) \geq \mu_i\), where
\[\mu_i = \max_{(A_1, A_2, \ldots, A_n) \in \mathcal{P}(M)} \min_{i \in N} u_i(A_i)\]

An allocation \(A\) is said to be \(\alpha\)-MMS if it guarantees \(u_i(A_i) \geq \alpha \cdot \mu_i\) for \(\mu_i \geq 0\), and \(u_i(A_i) \geq 1/\alpha \cdot \mu_i\) when \(\mu_i \leq 0\), where \(\alpha \in (0, 1]\).
We consider the following efficiency criteria.

Definition 7 (Pareto-Optimal (PO)). An allocation $A$ is PO if $\nexists A' s.t. \forall i \in N$, $u_i(A'_i) \geq u_i(A_i)$ and $\exists i \in N$, $u_i(A'_i) > u_i(A_i)$.

We consider Utilitarian welfare, for which an allocation $A$ is the sum of agents utilities. Similarly, Nash welfare corresponds to the product of agents’ utilities, and Egalitarian welfare corresponds to the minimum individual agents’ utility. Formally, the optimal welfare is,

Definition 8 (Maximum Welfare). The optimal welfare is,

- Maximum Utilitarian Welfare, $\MUW(u) = \max A \sum_{i=1}^{n} u_i(A_i)$ (7)
- Maximum Nash Welfare, $\MNW(u) = \max A \prod_{i=1}^{n} u_i(A_i)$ (8)
- Maximum Egalitarian Welfare, $\MEW(u) = \min A u_i(A_i)$ (9)

3 REDUCTION FROM 2-D TO 1-D

We define two transformations that convert 2-D utilities or valuations to corresponding 1-D valuations. We then represent the transformed valuations in 1-D as $W$. We denote the utility transformation as $(T_u)$ and valuation transformation as $(T_v)$.

Definition 9 (Transformation $(T_u)$). Given a resource allocation problem $(N, M, V)$ where $V$ represents 2-D valuations we obtain the 1-D valuations denoted by $W$ as follows, $\forall S \subseteq M$,

$$w_i(S) = u_i^W(S) = u_i^V(S) - v_i(M)$$ (10)

Each agent $i \in N$, has utility function $u_i^W \rightarrow \mathbb{R}_{\geq 0}$ for goods and $u_i^W \rightarrow \mathbb{R}_{\leq 0}$ for chores.

Given that $u_i^V$ is monotone, $u_i^W$ is normalized (i.e., $u_i(\emptyset) = 0$), monotone and non-negative for goods and accordingly for chores, i.e., anti-monotone and negative.

$T_u$ transforms utility from $V$ to utility in $W$. Now, in terms of valuations, the transformation that can reduce valuations from $V$ to $W$ for additive settings is known and is given below. However, such a transformation is unknown in general valuations.

Definition 10 (Transformation $(T_v)$). We consider the following valuation transformation from 2-D domain $V$ to a 1-D domain $W$. Each agent $i \in N$, has valuation function $w_i \rightarrow \mathbb{R}_{\geq 0}$ for goods and $w_i \rightarrow \mathbb{R}_{\leq 0}$ for chores. We define utility $u_i^W(S) = w_i(S)$

$$w_i(S) = v_i(S) - v_i^*(S)$$ (11)

Proposition 1. For additive valuations, utility transformation $(T_u)$ is equivalent to valuation transform $(T_v)$.

Proof Sketch.

$$u_i^W(S) = u_i^V(S) - v_i^*(M)$$

$$= v_i(S) + v_i^*(M \setminus S) - u_i^*(M)$$

$$= v_i(S) - v_i^*(S)$$

Similarly, we can prove vice versa. □

4 FAIRNESS IN 2-D DOMAIN

In this section, we analyze the challenges in adapting the existing fairness criteria and algorithms for the 2-D domain. We will explore how transformation can help us retain certain fairness properties, and thus we can transform into 1-D and apply the existing approaches. However, in the cases where transformation doesn’t work, we analyze their corresponding algorithms.

4.1 Analysis for EF and its relaxation

Proposition 2. Given a resource allocation problem $(N, M, V)$, an allocation $A$ is EF/EFX/EF1 in $u^V$, iff $A$ is EF/EFX/EF1 in $u^W$.

Proof Sketch. The utility transformation $T_u$ retains these properties.

$$\forall i \forall j, u_i^W(A_i) \geq u_i^W(A_j)$$

$$u_i^V(A_i) - v_i^*(M) \geq u_i^V(A_j) - v_i^*(M)$$

$$u_i^V(A_i) \geq u_i^V(A_j)$$

Similarly, the proof follows for EFX and EF1. Accordingly in additive setting $T_u$ retains these properties by Proposition 1. □

Given the above proposition, we make the following remarks for algorithms to find EF1 and EFX in additive and general 2-D domain.

Remark 1. EF1 for Additive Valuations: Round Robin for goods or chores and Double Round Robin for a combination [3, 16] find EF1 in polynomial time in 1-D domain. But $u^V$ is not additive; hence we cannot directly use the algorithms. We can apply the transformation $T_v$, i.e., we can transform 2-D domain to 1-D, $W$, and find EF1 allocations due to Proposition 2.

Remark 2. EF1 for General Valuations: Envy-cycle elimination algorithm [10, 32] finds EF1 allocations in 1-D general valuations. We can apply the utility transformation $T_u$, and apply envy-cycle elimination algorithm to $u^W$ to obtain EF1 allocation in $V$ due to Proposition 2.

Proof Sketch. Envy-cycle elimination algorithm is applicable for indivisible goods with general 1-D valuations. The assumption for the algorithm is that $\forall i \in N$, $v_i$ is non-negative, normalized, and monotone. In 1-D domain $\forall S \subseteq M$, $v_i = v_i(S)$, i.e., the utility function is non-negative, normalized, and monotone. Now, $u_i^W$ is non-negative, normalized, and monotone. We can apply the envy cycle elimination algorithm to $u^W$ to find EF1 allocation in $V$. Similarly, for chores presented in paper [10]. □

Remark 3. EFX for Identical valuations: Leximin++ finds EFX allocations for a 1-D domain with identical general valuations [34]. Leximin++ selects the allocation which maximizes the minimum individual utility. If there are multiple such allocations, it will further maximize the size of the bundle of the minimum utility, and then go on to maximizing the second minimum, and so on.

- Note that Leximin/Leximin++ allocation in 2-D domain and 1-D domain might not be same.

Example 1. Consider two agents $\{1, 2\}$ and two goods $\{g_1, g_2\}$. Agents have additive valuations. $V_1(g_1) = (8, -16)$, $V_1(g_2) = (10, -15)$, $V_2(g_1) = (5, -1)$, $V_2(g_2) = (6, -2)$, Leximin($V$) = $\{(g_1, g_2), (\emptyset)\}$, and Leximin($W$) = $\{(g_1), (g_2)\}$. 

Similarly, we can adapt the cut and choose algorithm for two agents where an agent makes a cut using Leximin++ solution, which gives EXF and PO.

For EF and its relaxation, we show how few of the current algorithms result in \(V\); however, given that the transformations retain this notion, we can adapt any algorithm via \(W\).

### 4.2 Analysis for EQ and its relaxations

**Proposition 3.** Given a resource allocation problem \((N, M, \mathcal{V})\),

1. With identical general valuations, an allocation \(A\) is EQ/EQX/EQ1 in \(u^W\) iff \(A\) is EQ/EQX/EQ1 in \(u^V\).
2. For non-identical valuations an allocation \(A\) is EQ/EQX/EQ1 in \(u^W\) does not necessarily mean \(A\) is EQ/EQX/EQ1 in \(u^V\), and vice versa, even for additive valuations.

**Proof Sketch a.** When the valuations are identical and additive EF/EFX/EF1 implies EQ/EQX/EQ1 [21]. Hence, from Proposition 2, we can prove the first statement.

**Proof Sketch b.** Consider the example, where \(N = \{1, 2\}\) and \(M = \{(g_1, g_2, g_3, g_4)\}\). The 2-D additive valuations for agent 1 for \(g_1 : (3, -6), g_2 : (3, -6), g_3 : (1, -3)\), and \(g_4 : (1, -3)\). For agent 2, the additive valuations for \(g_1 : (1, -8), g_2 : (1, -8), g_3 : (3, -6)\), and \(g_4 : (3, -6)\).

- \(A = \{(g_1, g_2), (g_3, g_4)\}\) is EQ in \(u^W\), but is not even EQ1 in \(u^V\).
- \(A = \{(g_3, g_4), (g_1, g_2)\}\) is EQ in \(u^V\) but is not even EQ1 \(u^W\).

Given the above proposition, applying \(T_{eq}\) to adapt algorithms proposed for 1-D is not possible unless identical valuations. Despite this, we make the following observation to ensure EQX.

**Remark 4.** EQX for Additive Valuations: Authors in [26] proposed a greedy algorithm to find EQX for goods when agents have 1-D additive valuations. We can apply this algorithm in \(V\).

**Proof Sketch.** \(\forall i \in N, u_j(A_i) \geq 1/n \sum_{j \in [n]} u_j(A_i) = 1/n \sum_{j \in [n]} u_j(M \setminus A_j)\) for additive valuations, \(1/n \sum_{j \in [n]} u_j(W(M))\) is equal to \(1/n \sum_{j \in [n]} u_j(V(M))\), and we get \(u_j(V(M)) \geq 1/n \sum_{j \in [n]} u_j(V(M))\). Hence A is EQX in \(V\) as given in Eq. 4. The proof for EQX and PROP-E will follow in similar fashion.

It is well known that in a 1-D domain, EXF implies EQX for sub-additive valuation. Whereas in a 2-D domain, EXF is always EQX, even for general arbitrary valuations, \(\forall i \in N, u_i(A_i) \geq u_i(M)\) for \(u_i(A_i) \geq 1/n \sum_{j \in [n]} u_j(A_i)\). Also EXF implies PROP-E and EQX in additive setting.

**Remark 5.** PROP-E and relaxations: Like PROP, PROP-E may not always exist, so we have relaxed versions and algorithms. As the transformation retains these properties, we can apply the algorithms in [5, 6] via \(W\).

### 4.3 Analysis for PROP-E

Ensuring PROP given by Eq. 3 is impossible in a 2-D domain; we use the definition given in Eq. 4. In [36] the authors have also modified PROP to include externalities when valuations are additive. An allocation is proportional when it ensures each agent the average-share. In the domain \(V\), the average value of item \(k\) for agent \(i \in N\), denoted by

\[
\text{avg}[v_{ik}] = 1/n \cdot [v_{ik}] + (n - 1)[v_{ik}'] \tag{12}
\]

The average-share of agent \(i \in N\),

\[
\bar{v}_i(M) = \sum_{k \in M} \text{avg}[v_{ik}] \tag{13}
\]

**Proposition 4.** For additive valuations, PROP-E reduces to Average-share, and both reduce to proportionality definition in a 1-D domain, i.e., no externalities.

**Proof Sketch.**

\[
\forall i \in N, u_i(A_i) \geq 1/n \sum_{j \in [n]} u_j(A_i) = 1/n \sum_{j \in [n]} u_j(M \setminus A_j) = 1/n \sum_{j \in [n]} v_{ik} - 1/n \sum_{j \in [n]} v_{j}'(M \setminus A_j) = 1/n \sum_{j \in [n]} (n - 1)v_{ik}' = \bar{v}_i(M)
\]

Similarly, we can prove versa.

**Proposition 5.** Given a resource allocation problem \((N, M, \mathcal{V})\), an allocation \(A\) is PROP-E/PROP-X/PROP-E1 in \(u^W\), iff \(A\) is PROP/PROP-E/PROP-E1 in \(u^W\) for additive valuations.

**Proof Sketch.** Consider an allocation \(A\) is PROP in \(u^W\), i.e., \(u_i^W(A_i) \geq 1/n \sum_{j \in [n]} u_j^W (A_j)\). For additive valuations, \(1/n \sum_{j \in [n]} u_j^W (A_j)\) is equal to \(1/n \sum_{j \in [n]} u_j^W (M)\). Now, \(u_j^W (A_j) + \psi_j^W (M) \geq 1/n \sum_{j \in [n]} u_j^W (A_j) + \psi_j^W (M)\), and we get \(u_j^W (A_j) \geq 1/n \sum_{j \in [n]} u_j^W (A_j)\). Hence A is PROP-E in \(V\) as given in Eq. 4. The proof for PROP-E and PROP-E1 will follow in similar fashion.

It is well known that in a 1-D domain, EXF implies PROP for sub-additive valuation. Whereas in a 2-D domain, EXF is always PROP-E, even for general arbitrary valuations, \(\forall i \in N, u_i(A_i) \geq u_i(M)\) for \(u_i(A_i) \geq 1/n \sum_{j \in [n]} u_j(A_i)\). Also EXF implies PROP-E and EQX in additive setting.

**Remark 5.** PROP-E and relaxations: Like PROP, PROP-E may not always exist, so we have relaxed versions and algorithms. As the transformation retains these properties, we can apply the algorithms in [5, 6] via \(W\).

### 4.4 Analysis for MMS

**Proposition 6.** Given a resource allocation problem \((N, M, \mathcal{V})\), an allocation \(A\) is MMS in \(u^W\), iff \(A\) is MMS in \(u^W\)

**Proof Sketch a.** Proof follows similar to Proposition 2, resulting into

\[
u_i^W (A_i) - v_i^W (M) \geq -v_i^W (M) + \max_{A \in \mathcal{A}_i(M)} \min_{j \in N} u_j^W (A_j)
\]

□
MMS allocation doesn’t always exists, and even if it does computing it is NP-hard, but a PTAS exists [40]. There are various algorithms for approximate MMS or α-MMS [1, 7, 8, 23–25, 27, 35]. Unfortunately, α-MMS is not retained through $T_\alpha$. We know that $\mu_i$ is positive for goods and negative for chores in a 1-D domain. However, we can no longer guarantee that for goods with negative externalities and chores with positive externalities. For a resource allocation problem, some agents can have $\mu_i \geq 0$, while others can have $\mu_i \leq 0$. We consider the generalized definition of $\alpha$-MMS as defined in Def. 6.

**Proposition 7.** Given a resource allocation problem $(N, M, V)$,

- If an allocation $A$ is $\alpha$-MMS in $u^W$ then $A$ is $\alpha$-MMS in $u^V$, for positive externalities in goods, and negative externalities in chores but not vice versa.
- If an allocation $A$ is $\alpha$-MMS in $u^W$ then $A$ is $\alpha$-MMS in $u^V$, for negative externalities in goods, and for positive externalities in chores but not vice versa.

**Proof Sketch [a].** If allocation $A$ is $\alpha$-MMS in $W$, then

$$\forall i \in N, u_i^W(A_i) \geq \alpha \cdot \mu_i^W$$

$$u_i^V(A_i) = c_i'(M) \geq c_i'(2^{\alpha \cdot \mu_i^V})$$

So in the case of positive externalities for goods, $\alpha \in [0,1)$, and $V \subseteq M$, $\gamma'(S) \geq 0$, and in the case of negative externalities for chores, $\alpha \geq 1$, and $V \subseteq M$, $\gamma'(S) \leq 0$, and so $c_i'(M) \leq \alpha \cdot c_i'(M)$

So we can use the algorithm in the 2-D domain to obtain $\alpha$-MMS. We make the following remarks towards some negative results along these lines.

**Example 2.** Consider $N = \{1, 2\}$ and $M = \{g_1, g_2, g_3, g_4, g_5, g_6\}$, both having additive identical valuations, given by, $g_1 : (0, 5, 0, 1)$, $g_2 : (0, 5, 0, 1), g_3 : (0, 5, 0, 1), g_4 : (0, 5, 0, 1)$, and $g_5 : (0, 5, 0, 1), g_6 : (0, 5, 0, 1)$. $\mu_i^W = 1$ and $\mu_i^V = 1.6$. In $V$, $A = \{(g_1), (g_2, g_3, g_4, g_5, g_6)\}$ is 1/2~MMS, while it is not in $W$.

**Proof Sketch [b].** If allocation $A$ is $\alpha$-MMS in $V$, then there are two conditions. Consider in the case of goods in $V$, $i, \mu_i \geq 0$; $\mu_i < 0$.

We will denote $\alpha_i'$ for $V$. Note that $\alpha_i' \in [0, 1)$ in $W$.

If allocation $A$ is $\alpha$-MMS in $W$, then

$$\forall i \in N, u_i^W(A_i) \geq \alpha_i' \cdot \mu_i^W$$

$$u_i^V(A_i) = c_i'(M) \geq c_i'(1/\alpha_i') \cdot \mu_i^V$$

- $\alpha_i' = 1/\alpha_i W > 1$, i.e., $\mu_i < 0$

$$u_i^W(A_i) + c_i'(M) \geq 1/\alpha_i' \cdot c_i'(M) + 1/\alpha_i W \cdot \mu_i^W \geq \alpha_i' \cdot c_i'(M) + \alpha_i W \cdot \mu_i^V$$

$$Now, c_i'(M) \leq \alpha_i W \cdot c_i'(M) \implies u_i^W(A_i) \geq \alpha_i W \cdot \mu_i^W$$

- $\alpha_i' = \alpha_i W \in [0, 1)$, i.e., $\mu_i > 0$ The proof follows similarly. However, if allocation $A$ is $\alpha$-MMS in $W$, does not mean it is $\alpha$-MMS in $V$.

**Example 3.** Consider $N = \{1, 2\}$ and $M = \{g_1, g_2, g_3, g_4, g_5, g_6, g_7\}$, both having additive identical valuations, given by, $g_1 : (0.1, -0.2), g_2 : (0.2, -0.1), g_3 : (0.1, -0.1), g_4 : (0.1, -0.3), g_5 : (0.1, -0.3), g_6 : (0.1, -0.3)$ and $g_7 : (0.1, -0.3), \mu_i^W = -0.2$ and $\mu_i^V = 1$ Allocation $A = \{(g_1, g_2), (g_3, g_4, g_5, g_6, g_7)\}$ is 1/2~MMS in $W$, while it is not 2~MMS in $V$.

The proof follows similarly for chores.

**Proposition 8.** Bag Filling does not guarantee $(1 - \delta)$-MMS allocation for goods with negative externalities when applied in $V$.

**Proof.**

**Example 4.** Consider 3 agents - $N = \{1, 2, 3\}$, having additive valuations for 12 goods - $M = \{i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9, i_{10}, i_{11}, i_{12}\}$ provided in Table 1. The valuation profile in Table 1 is based on the MMS doesn’t exist example in the paper [30]. For calculating MMS Share, Agent 1 divides items as $(\{i_1, i_2, i_7, i_{11}\}), (i_{10}, i_4, i_9, i_1), (i_8, i_2, i_5, i_11))$, Agent 2 divides items as $(\{i_6, i_10, i_4, i_2\}), (i_7, i_4, i_2, i_{12}), (i_1, i_8, i_1, i_5))$, and Agent 3 divides items as $(\{i_5, i_6, i_3, i_2\}), (i_{12}, i_{10}, i_1, i_{12}), (i_{11}, i_4, i_8, i_5))$.

In Bag Filling, similar to the famous moving knife algorithm, for goods valued $\leq \delta \mu_i$, we keep adding goods in a bag until some agent claims it, i.e., the agent values it at least $(1 - \delta) \mu_i$, and so on. We set $\delta = 1/2$, i.e., 1/2-MMS allocation. In Table 1, for $W$, $\forall i, \mu_i^W = 950, \mu_i^V = 1$ $\forall i, \delta \mu_i \leq 475$, and hence we can apply bag filling to obtain 1/2-MMS allocation. Note that in $V$, we start filling the bag, till $(g_1, g_2)$, no one claims it. As the bag reaches $(g_2, g_3, g_4)$, all three claim it, and we break ties lexicographically and assign it to agent 1, $A_1 = \{g_1, g_2, g_3\}$, and $u_i^W(A_1) = 110$. We again start filling till $(g_4, g_5, g_6)$, no one claims it. As the bag reaches $(g_4, g_5, g_6, g_7)$, both agents 2 and 3 claims it. We assign the

| Allocation | $V$ | $W$ | $\mathbf{w}$ |
|------------|-----|-----|-------------|
| $(g_1, g_2)$ | $(g_1, g_2)$ | $(g_1, g_2)$ | $(g_1, g_2)$ |
| $(g_3, g_4, g_5, g_6, g_7)$ | $(g_1, g_2)$ | $(g_1, g_2)$ | $(g_1, g_2)$ |
| $(g_8, g_9, g_{10}, g_{11}, g_{12})$ | $(g_1, g_2)$ | $(g_1, g_2)$ | $(g_1, g_2)$ |
| $(g_{13}, g_{14}, g_{15}, g_{16}, g_{17})$ | $(g_1, g_2)$ | $(g_1, g_2)$ | $(g_1, g_2)$ |
| $(g_{18}, g_{19}, g_{20}, g_{21}, g_{22})$ | $(g_1, g_2)$ | $(g_1, g_2)$ | $(g_1, g_2)$ |
| $(g_{23}, g_{24}, g_{25}, g_{26}, g_{27})$ | $(g_1, g_2)$ | $(g_1, g_2)$ | $(g_1, g_2)$ |

Table 1: Additive Identical Ordinal Valuation Profile
bag to agent 2, $A_2 = \{g_4, g_5, g_6, g_7\}$, and $u^W_1(A_2) = 63$. and we assign remaining items to agent 3, $A_3 = \{g_8, g_9, g_{10}, g_{11}, g_{12}\}$, and $u^W_3(A_3) = -280$. This allocation is not 1/2-MMS in $\mathcal{V}$ i.e., in 2-D domain proving our proposition.

Further in $\mathcal{W}$, the allocation found using bag filling $A = \{(g_1, g_2), (g_3, g_4), (g_5, g_6, g_7, g_8, g_9, g_{10}, g_{11}, g_{12})\}$ is 1/2-MMS in $\mathcal{W}$.

**Remark 7.** In the 1-D domain, Bag Filling works for any additive valuations; however, algorithms in [23, 24] first converted these valuations into ordered instances and then apply bag filling for low-valued items. However, in 2-D domain, Bag filling doesn’t work even for IDO (Identical Ordinal Preference) instance

**Remark 8.** For chores, authors in [7] presented a Round Robin algorithm for 2-MMS allocation, and authors in [27] presented an algorithm that gives 11/9-MMS. However, both these algorithms don’t work as in $\mathcal{V}$

**Proposition 9.** For chores with positive externalities in $\mathcal{V}$, Round Robin allocation doesn’t guarantee 2-MMS allocation.

**Proof.** Consider the following example,

**Example 5.** Consider 2 agents - $N = \{1, 2\}$ and 3 chores - $M = \{c_1, c_2, c_3\}$, both having additive identical valuations, given by, $c_1: (-40, 36)$, $c_2: (-39, 36)$, and $c_3: (-120, 60)$. $\mu^W_1 = -48$, and $\mu^W_2 = -180$. In Round Robin, the agents arrive in a fixed order and pick the item with the maximum utility of the remaining items sequentially. Agent 1 choose first and Round Robin allocation is $A = \{(c_2, c_3), (c_1)\}$. A is 2-MMS in $\mathcal{W}$, but not in $\mathcal{V}$

**Proposition 10.** For chores with positive externalities in $\mathcal{V}$, the algorithm in [27] doesn’t guarantee 11/9-MMS allocation.

**Proof.** Consider the following example,

**Example 6.** Consider 3 agents - $N = \{1, 2, 3\}$, having additive valuations for 12 chores - $M = \{i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9, i_{10}, i_{11}, i_{12}\}$. In Table 1, we negated all the valuations, i.e., $v_{11} = -256$, $v_{11} = 75$, and $w_{11} = -380$. Now $v_i, i = 1, 2, \ldots, 12, \mu^W_1 = -950$, and $\mu^W_2 = -1$. We set a threshold value for each agent, and then iterate $n$ times to create $n$ bundles of chores to allocate among agents. In each iteration, we add chores from largest (more negative) to lowest until, $\exists i, a_i = a_i$, we allocate the bundle to $i$ and repeat the process with remaining agents and chores. We set the threshold value to 11/9 of MMS for each agent, $v_i, i = 1, 2, \ldots, 12$, and $\mu^W_2 = -1161.112$.

In $\mathcal{W}$, we add chores $\{i_1, i_2\}$ to a bundle; it doesn’t violate any agent’s constraint. Now, as we add chore $i_3$, it violates all three agents’ threshold value. Similarly, adding a chore from $\{i_3, i_4, i_5\}$ to this bundle will again violate. On adding $i_7$, the bundle $\{i_1, i_2, i_7\}$ satisfies $\mathcal{W}$, and further adding any chore will only violate. We break ties lexicographically and assign the bundle to agent 1, $A_1 = \{i_1, i_2, i_7\}$, $\mathcal{W}$ = 1. Similarly, we continue with the algorithm, and we get $A_2 = \{i_3, i_4, i_6\}$, $A_3 = \{i_5, i_7, i_8, i_9\}$, leaving chores $\{i_{10}, i_{11}, i_{12}\}$ unallocated. Assigning them to any agent will violate 11/9-MMS. On apply this in $\mathcal{W}$, we get $A = \{(i_1, i_2, i_3, i_4), (i_5, i_6, i_7, i_8), (i_9, i_{10}, i_{11}, i_{12})\}$ which is 11/9-MMS in $\mathcal{W}$.

We cannot retain $\alpha$-MMS via transformation for goods with negative externality and chores with positive externality, and we cannot apply the algorithm presented in [7, 23–25, 27] in $\mathcal{W}$. It leaves further scope to explore the algorithms for the same.

## 5 Efficiency in 2-D Domain

We will now analyze if the transformation defined in Equation 10 also helps us find efficient allocations in the 2-D domain.

We find utility transformation holds for PO and MUW and does not hold for MEW and MNW. MNW cannot be defined in $\mathcal{V}$ as some agents might have non-negative utilities while some might have negative utilities.

**Proposition 11.** Given a resource allocation problem $(N, M, \mathcal{V})$, an allocation $A$ is PO in $\mathcal{V}$, if $A$ is PO in $\mathcal{W}$

**Proposition 12.** Given a resource allocation problem $(N, M, \mathcal{V})$, an allocation $A$ is MUW in $\mathcal{V}$, if $A$ is MUW in $\mathcal{W}$

Proof for Proposition [11,12] follows in similar fashion.

**Proposition 13.** Given a resource allocation problem $(N, M, \mathcal{V})$ an allocation $A$ is MEW in $\mathcal{V}$ does not necessarily mean $A$ is MEW in $\mathcal{W}$, even with additive valuations.

Consider valuation profile in Example 1, $\mathcal{MEW}(\mathcal{V}) = \{(g_1, g_2), (\phi), (\phi)\}$, while $\mathcal{MEW}(\mathcal{W}) = \{(g_1), (g_2)\}$.

**Remark 9.** If we can retain certain fairness and efficiency properties through transformation, we can adapt all the corresponding algorithms to ensure fair and efficient allocations. Few examples are as follows:

- The authors [16] proved that MNW is EF1 and PO for goods. In our setting, we cannot define Nash welfare as described in Section 5. Hence we cannot calculate MNW directly on our 2-D domain. However, given the Proposition 2, we can reduce our valuations to 1-D and then find MNW to get EF1 and PO allocations. Similarly, the algorithm in [9], we can reduce our valuations to get EF1 and PO. The authors in [3] introduced Winner Adjusted Rule for calculating EF1 and PO for $n = 2$ for goods, chores, or mixed. Similarly, we can reduce to $\mathcal{W}$ and can apply these algorithms directly.

- The paper [6] presented an algorithm to find fractional PO which is stronger than PO and PROP1. Given Proposition 5, we can find PO and PROP1-E by reducing our $\mathcal{W}$.

## 6 Discussion

In this paper, we conducted a study on indivisible resource allocation with externalities, specifically in the 2-D domain. We proposed a transformation from the 2-D domain to the 1-D domain, studied how we retained fairness/efficiency notions, and analyzed the existing algorithms. We observed that $\alpha$-MMS, EQ, and MEW require more analysis in this setting, as we couldn’t retain these notions using transformation. Also, we studied over a 2-D domain, leaving scope to explore general externalities.

**References**

[1] Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. 2017. Approximation Algorithms for Computing Maximin Share Allocations. ACM Trans. Algorithms 13, 4, Article 52 (Dec. 2017), 28 pages. https://doi.org/10.1145/3147173
