Maximin share allocation for two types of items

Yu Chen, **Kai Jie Wong** and **Xiaorui Zhang**
National University of Singapore

Abstract

We study the fair allocation of indivisible items among n agents with additive valuations (or costs) using the fairness notion of maximin share (MMS). Since MMS allocations do not always exist when $n > 2$, one direction in the literature has been to identify special classes of instances in which exact MMS allocations can be guaranteed. We show that, when items can be divided into two types of chores, MMS allocations always exist and can be computed in polynomial time using a geometric approach inspired by prior work in job scheduling. We further discuss how our technique extends to instances with two types of goods, as well as mixed instances consisting of one type of objective good and one type of objective chore.

1 Introduction

The fair allocation of indivisible items among agents with differing preferences arises naturally in many settings and has attracted significant interest across disciplines, such as in multi-agent systems [Aziz, 2020]; see also recent surveys [Amanatidis *et al.*, 2023; Liu *et al.*, 2024]. The goal is to find a *fair* partition of a set M of m items among a set N of n agents, where each agent $i \in N$ has some valuation function $v_i(\cdot)$ over subsets of M. In this work, we assume that agents have additive preferences, such that the utility of a bundle S is given by $v_i(S) = \sum_{o \in S} v_i(o)$, where $v_i(o)$ represents the utility of item o. An item o can be considered to be a *good* (in which case $v_i(o) > 0$, or a *chore* ($v_i(o) < 0$).

One natural and widely-studied fairness notion in this setting is the *maximin share* (MMS) value. Introduced by Budish [2011] in the context of course allocation, an agent's MMS value is the maximum utility they can guarantee for themselves, if they partition the items into n bundles, then receive the worst one. Formally, let Π denote the set of all partitions of M into n bundles. Let $(X_i)_{i \in N} \in \Pi$ be any such partition. Then, we write the MMS value for agent $i \in N$ as

$$\text{MMS}_i := \max_{(X_i)_{i \in N} \in \Pi} \min_{j \in N} v_i(X_j).$$

An allocation is said to be MMS if every agent receives a bundle worth at least their MMS value.

The MMS is a compelling benchmark for fairness, with evidence from real-world experiments indicating that participants tend to prefer it [Gates *et al.*, 2020]. However, despite its intuitive appeal, exact MMS allocations may not always exist, even when there are only $n \geq 3$ agents with additive valuations [Kurokawa *et al.*, 2018; Feige *et al.*, 2021].

In view of the difficulty of attaining MMS allocations for general additive valuations, we turn our attention to the special case when there are at most two types of items. We assume that the item set M can be partitioned into two "types" A and B. Items of the same type are identical, so a fixed agent will derive the same utility from items of the same type. A motivating example, in the case of chores, could be a group of housemates fairly splitting up $|A|$ cooking chores, and $|B|$ cleaning chores.

At the same time, we consider the more general case of mixed instances, where items may be either goods or chores. For our setting, we assume that the items are either *objective goods* (which yield positive utility for all agents) or *objective chores* (which yield negative utility for all agents), as formalized by Aziz et al. [2021].

1.1 Contributions

We give a polynomial-time algorithm for computing MMS allocations for two-item-type instances in each of the following settings:

- at most two types of chores
- at most two types of goods
- exactly one type of *objective good* and one type of *objective chore* (a "mixed" instance).

In the case of chore allocation, it is well-known that when agents are identical, the MMS allocation problem is equivalent to the classic makespan minimization in job scheduling [Huang and Segal-Halevi, 2023]. Building on this connection, our approach is based on a geometric perspective, inspired by prior work in job scheduling when there are only two types of jobs [McCormick *et al.*, 1997].

A similar geometric viewpoint was also used by Gorantla et al. [2023] to obtain EFX (envy-free up to any good) allocations for two types of goods. However, their techniques differ from ours due to the different nature of EFX and MMS.

Since there are few settings in which MMS allocations are known to exist, these results provide new positive cases for exact MMS computability. A notable feature of our approach is that essentially the same underlying technique applies across goods, chores, and mixed instances.

We first present the algorithm and proof for the chore setting in Sections 2 and 3. In Section 4, we show how the same ideas extend naturally to goods and mixed instances using analogous arguments.

1.2 Further related work

Few types of items. Several works have studied this setting in the case of envy-based fairness, under additive valuations. Aziz et al. [2023] provide a polynomial-time algorithm for EFX allocations when there are only two types of chores. Gorantla et al. [2023] give a polynomial-time algorithm for EFX allocations where there are only two types of goods, as well as existence of envy-free allocations for arbitrary number of types of goods, provided each type appears in sufficiently large quantity.

Other special cases. The two-item-types setting can also be viewed as a natural subclass of *personalised bivalued* instances[1], for which the best-known approximation factor for chores is $\frac{15}{13}$ due to Garg et al. [2025] and existence remains unknown. For the case of (non-personalized) bivalued instances[2], MMS allocations are known to exist [Feige, 2022], for both goods and chores. There is also a line of work showing how MMS allocations can be computed in polynomial time when agents have binary submodular valuations for goods [Barman and Verma, 2021; Viswanathan and Zick, 2025], or binary supermodular costs for chores [Barman *et al.*, 2023].

Approximations. Since MMS allocations are not guaranteed in general additive instances, the study of approximate MMS allocations has also seen significant progress. For goods, the recent work of Huang and Zhou [2025] achieve the best known approximation factor of $\frac{7}{9}$ under additive valuations. For chores, Huang and Segal-Halevi [2023] achieve the best known approximation factor of $\frac{13}{11}$ under additive disutilities via a reduction to job scheduling.

2 Preliminaries

Let M be a set of m indivisible chores, and N be a set of n agents. In this section and Section 3, we focus on chore allocation. Since all items yield negative utility, it is more convenient to work with positive disutilities instead. Accordingly, for each agent $i \in N$ we let $d_i : M \to \mathbb{R}_{>0}$ denote an additive disutility function. We consider the setting where the chore set can be partitioned as $M = A \cup B$ where for each $i \in N$, if $c \in A$ then $d_i(c) := \alpha_i$ for some $\alpha_i \in \mathbb{R}_{>0}$, and if $c \in B$ then $d_i(c) := \beta_i$ for some $\beta_i \in \mathbb{R}_{>0}$. If $c \in A$, we say that c is a Type A chore; otherwise it is a Type B chore. Since chores of the same type are identical, any two-chore-type fair division instance can be succinctly described by $\mathcal{I} := (a, b, N)$ where a, b are the number of Type A and Type B chores respectively.

Maximin share. Recall Π is the set of all partitions of M into n bundles. For an instance of chore allocation, we instead define the MMS share as a positive value

$$\mathrm{MMS}_i(M) := \min_{(X_i)_{i\in N}\in\Pi} \max_{j\in N} d_i(X_j),$$

thus an allocation is said to MMS if every agent receives a bundle of disutility at most their MMS value. For convenience, we also introduce a more general notion:

D-feasibility. For some threshold $t \in \mathbb{R}$, a bundle B is *t-feasible* for agent $i \in N$ if $d_i(B) \le t$. We say that an allocation $X = (X_1, \dots, X_n)$ is *t-feasible* for agent $i \in N$ if for all $j \in N, d_i(X_j) \le t$. Let $D = (D_1, \dots, D_n) \in \mathbb{R}^n$ be a vector of thresholds. We say that an allocation X is *D-feasible* if it is D_i-feasible for all $i \in N$.

The MMS allocation problem is the special case where $D_i = \mathrm{MMS}_i$ for all $i \in N$.

2.1 Geometric formulation

When there are only two types of chores, any bundle S assigned to an agent can be represented by an integral pair $(x, y) \in \mathbb{Z}^2_{\ge 0}$ where x is the number of Type A chores and y is the number of Type B so that the disutility of the bundle for agent i is $d_i(S) = \alpha_i x + \beta_i y$. Observe that the points $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \in \mathbb{Z}^2_{\ge 0}$ correspond to a complete allocation if $\sum_{i=1}^n (x_i, y_i) = (a, b)$, i.e. the allocation consists of nonnegative integral bundles which uses up all a Type A chores and all b Type B chores. From now on we may use the terms "integer point" and "bundle" interchangeably.

For $t \in \mathbb{R}$, define $L_i(t)$ to be the line $\alpha_i x + \beta_i y = t$. The set of (possibly fractional) t-feasible bundles for agent i is exactly the nonnegative region $R_i(t)$ defined by

$$R_i(t) := \{(x, y) \in \mathbb{R}^2_{\ge 0} : \alpha_i x + \beta_i y \le t\},$$

which is also known as the *fractional knapsack polytope*. Let the convex hull of integral points (the "integer hull") in $R_i(t)$ be $IH_i(t) := \mathrm{conv}(R_i(t) \cap \mathbb{Z}^2)$. This is the corresponding *knapsack polytope*.

Given a two-chore-type instance $\mathcal{I} = (a, b, N)$, we call the point $P_{\mathcal{I}} := (\frac{a}{n}, \frac{b}{n})$ the *feasibility point* of the instance. We have the following useful observation.

We write '+' as regular vector addition of points, and for any point T, we write $x(T)$ and $y(T)$ to be its x- and y-coordinates respectively.

Observation 1. *$P_{\mathcal{I}} = (\frac{a}{n}, \frac{b}{n}) \in \cap_{i=1}^n IH_i(MMS_i)$ for each $i \in N$.*

Proof. Observe that given $\mathcal{I}$, any MMS allocation for agent $i \in N$ corresponds to n points $(x_1, y_1), \dots, (x_n, y_n) \in R_i(\mathrm{MMS}_i) \cap \mathbb{Z}^2$ such that $\sum_{i=1}^n (x_i, y_i) = (a, b)$. Then, by convexity, their centroid P lies in $IH_i(\mathrm{MMS}_i)$. □

2.2 Unimodular triangle allocations

Next, we adapt results from McCormick et al. [1997], which characterizes the feasibility of two-job-type job scheduling allocations based on the feasibility point.

[1] An instance is *personalised bivalued* if $v_i(o) \in \{a_i, b_i\}$ for all $o \in M$.

[2] An instance is *bivalued* if $v_i(o) \in \{a, b\}$ for all $o \in M$.

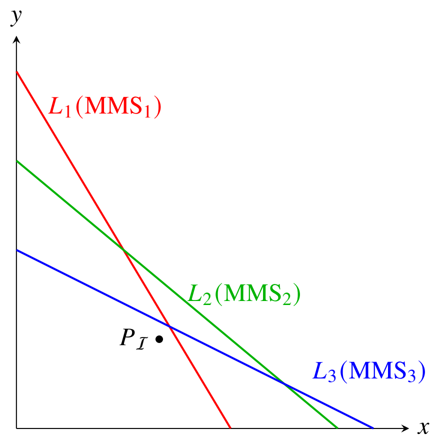
We restate the result (Corollary 3.6 in McCormick et al. [1997]) in our setting. For completeness, we also include a proof.

Lemma 1. *For any two-chore-type instance $\mathcal{I} = (a, b, N)$, fix some threshold $t \in \mathbb{R}$. For each $i \in N$, the following statements are equivalent.*

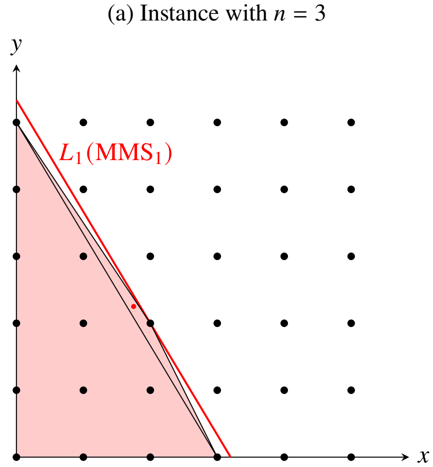
(i) $P_{\mathcal{I}} \in IH_i(t)$

(ii) $P_{\mathcal{I}}$ *lies in some unimodular (area 1/2) triangle with vertices in* $R_i(t) \cap \mathbb{Z}^2$

(iii) There is a triplet of integral points $(x_1, y_1), (x_2, y_2), (x_3, y_3) \in R_i(t) \cap \mathbb{Z}^2$ *and integral multiplicities* $n_1, n_2, n_3 \in \mathbb{Z}_{\geq 0}$ *such that* $n_1(x_1, y_1) + n_2(x_2, y_2) + n_3(x_3, y_3) = (a, b)$ *and* $n_1 + n_2 + n_3 = n$.



(a) Instance with $n = 3$



(b) $L_1(\text{MMS}_1)$ is $5x + 3y = 16$. $P_{\mathcal{I}} = (\frac{7}{4}, \frac{9}{4}) = \frac{1}{4}((0,5) + 2 \cdot (2,2) + (3,0))$. Shaded region is $IH_1 = \text{conv}(R_1 \cap \mathbb{Z}^2)$.

Figure 1: An illustration of Observation 1 and Lemma 1

Proof. (i) $\implies$ (ii): Observe that $IH_i(t)$ is a lattice polygon which can be triangulated such that no triangle contains a lattice point within its interior (to see this, if any triangle contains an interior lattice point, connect that point to the three vertices. If any edge contains additional lattice points, subdivide along these points. This refinement must terminate). By Pick's theorem these triangles have area 1/2.

(ii) $\implies$ (iii): Suppose $P_{\mathcal{I}}$ lies within the unimodular triangle T with vertices $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. Define

$$W := \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix}$$

The area of the triangle T is given by $\frac{1}{2}|\det W|$. Since T has area $\frac{1}{2}$ by definition, then $\det W = \pm 1$, i.e. W is a unimodular matrix.

Since $P_{\mathcal{I}} = (\frac{a}{n}, \frac{b}{n})$ lies within T, for some $\lambda_1, \lambda_2, \lambda_3 \in \mathbb{R}_{\geq 0}$, we can write

$$W \cdot \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} a/n \\ b/n \\ 1 \end{bmatrix} \implies W \cdot \begin{bmatrix} n\lambda_1 \\ n\lambda_2 \\ n\lambda_3 \end{bmatrix} = \begin{bmatrix} a \\ b \\ n \end{bmatrix}.$$

Now, since we know that W has integral entries, $\det W = \pm 1$, and a, b, n are integers, then by applying Cramer's Rule, $n\lambda_1, n\lambda_2, n\lambda_3$ are integral. Then set $n_1 := n\lambda_1, n_2 := n\lambda_2, n_3 := n\lambda_3$.

(iii) $\implies$ (i): By definition, each $(x_j, y_j) \in R_i(t) \cap \mathbb{Z}^2$, and $\sum_{i=1}^{n}(x_j, y_j) = (a, b)$. Then, by convexity, their centroid $P_{\mathcal{I}}(x/N, y/N)$ lies in $IH_i(t)$. □

Observation 1 along with Lemma 1 implies that for any two-chore-type instance $\mathcal{I}$, for any $t \geq \text{MMS}_i$, there exists a t-feasible allocation for each agent i which uses at most three different types of bundles of integral multiplicities $n_1 + n_2 + n_3 = n$. These bundles form a unimodular triangle with weighted centroid $P_{\mathcal{I}}$. By Lemma 3.3 of McCormick et al. [1997], this triangle can be found in polynomial time:

Lemma 2. *Given a two-chore-type instance $\mathcal{I}$, an agent i, and a threshold $t \geq MMS_i$, a unimodular triangle with vertices in $R_i(t) \cap \mathbb{Z}^2$ containing $P_{\mathcal{I}}$ exists and can be found in $O(\log^2 t)$ time.*

For our main result, we make use of this algorithm as a subroutine. Given two-chore-type instance $\mathcal{I}$ and threshold t, we denote by $\triangle_i(\mathcal{I}, t)$ the triangle computed by Lemma 2, and by $\mu_i(\mathcal{I}, t)$ the corresponding multiplicities of its vertices.

3 MMS allocation for two types of chores

In this section, we describe the Allocate procedure (Algorithm 1) and prove the following theorem, which implies the existence of exact MMS allocations when there are only two types of chores by setting $(D_i)_{i \in N} := (\text{MMS}_i)_{i \in N}$ then applying Observation 1.

Theorem 1. *Given a two-chore-type instance $\mathcal{I} = (a, b, N)$, fix some threshold vector $D = (D_i)_{i \in N} \in \mathbb{R}^n$. Suppose $P_{\mathcal{I}} \in \cap_{i \in N} IH_i(D_i)$. Then Allocate$(\mathcal{I}, D)$ (Algorithm 1) returns a D-feasible allocation in polynomial time.*

Proof. Throughout the algorithm, we say that a point X is "top-right" of $P_{\mathcal{I}}$ if $x(X) \geq x(P_{\mathcal{I}})$ and $y(X) \geq y(P_{\mathcal{I}})$. The terms "top-left", "bottom-left", "bottom-right" are also defined accordingly.

The algorithm works recursively. The idea is that in each iteration, we try to assign feasible bundles to some subset of agents. Suppose we are left with a remaining instance $\mathcal{I}'$ consisting of $a' \leq a$ Type A chores, $b' \leq b$ Type B chores, and a set N' of $n' < n$ agents. Then if we can show that the new feasibility point $P_{\mathcal{I}'} = (\frac{a'}{n'}, \frac{b'}{n'})$ remains in $IH_i(D_i)$ for each $i \in N'$, then we are done (by induction).

First we define a useful partial order. Let v_i be the value of y where $\alpha_i \cdot x(P_{\mathcal{I}}) + \beta_i \cdot y = D_i$, and let h_i be the value of x where $\alpha_i \cdot x + \beta_i \cdot y(P_{\mathcal{I}}) = D_i$. Define a partial order $\preceq$ on the agents N so that for $i, j \in N$,

$$i \preceq j \iff v_i \leq v_j \text{ and } h_i \leq h_j.$$

Algorithm 1 fixes $i \in N$ to be a minimal agent under $\preceq$ (see Fig 2). Intuitively, L_i will be "one of the closest" to $P_{\mathcal{I}}$ among all the lines $(L_j)_{j \in N}$.

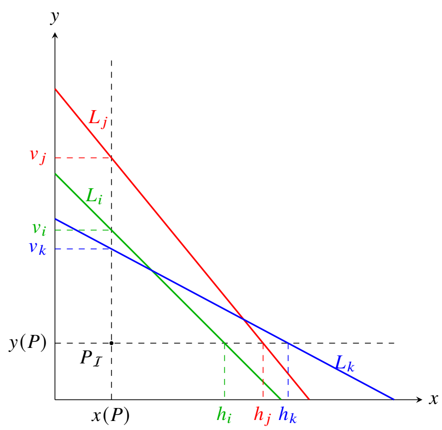


Figure 2: $L_i \preceq L_j$. (L_i, L_j) and (L_j, L_k) are incomparable. If $x < x(P)$, then L_i is under L_j. If $y < y(P)$, then L_i is under L_k.

Depending on the geometric configuration of vertices of $\triangle_i(\mathcal{I}, D_i)$ (i.e. whether they lie "top-left"/"bottom-left"/etc. of $P_{\mathcal{I}}$) and their corresponding multiplicities, Algorithm 1 assigns these bundles to different agents case-by-case. We prove the correctness of each case separately.

Given some threshold vector $D = (D_i)_{i \in N} \in \mathbb{R}^n$ as input, for $i \in N$ we write $L_i, R_i, IH_i, \triangle_i(\mathcal{I}), \mu_i(\mathcal{I})$ to mean $L_i(D_i), R_i(D_i), IH_i(D_i), \triangle_i(\mathcal{I}, D_i), \mu_i(\mathcal{I}, D_i)$ respectively.

If $n = 1$, then there is a single agent $i \in N$ with $P_{\mathcal{I}} \in IH_i$. Then we may simply assign the bundle P to agent i since it is D_i-feasible for them. Assume the claim holds for all $n' < n$, and suppose $n > 1$.

Let $i \in N$ be a minimal agent under $\preceq$ as fixed by Algorithm 1. Now let A, B, C be the vertices of $\triangle_i(\mathcal{I})$, and $n_A, n_B, n_C \in \mu_i(\mathcal{I})$ their corresponding multiplicities.

Case 1: Some $X \in \{A, B, C\}$ lies "top-right" of $P_{\mathcal{I}}$.

See Fig 3. Since $X \in R_i(D_i) \cap \mathbb{Z}^2$, it is D_i-feasible for agent i so we may assign them bundle X. For the remaining instance $\mathcal{I}'$, the new feasibility point $P_{\mathcal{I}'} = (\frac{a - x(X)}{n-1}, \frac{b - y(X)}{n-1})$ will lie away from $P_{\mathcal{I}}$ in the direction of $\overrightarrow{XP_{\mathcal{I}}}$, which is "bottom-left" of $P_{\mathcal{I}}$. Since $a - x(X) \geq 0$ and $b - y(X) \geq 0$, $P_{\mathcal{I}'}$ lies within the rectangle defined by the points

---

Algorithm 1 $\textsc{Allocate}(\mathcal{I}, D)$

Require: Instance $\mathcal{I} = (a, b, N)$ and threshold vector $D = (D_1, \ldots, D_n)$

if $n = 1$ **then return** Assign $P_{\mathcal{I}}$ to the single agent
Fix a minimal agent $i \in N$ under $\preceq$
Let A, B, C be the vertices of $\triangle_i(\mathcal{I}, D_i)$, and $n_A, n_B, n_C \in \mu_i(\mathcal{I}, D_i)$ their corresponding multiplicities.
if $\exists X \in \{A, B, C\}$ "top-right" of $P_{\mathcal{I}}$ **then** ▷ Case 1
  Assign X to agent i
  return $\textsc{Allocate}(a - x(X), b - y(X), N \setminus \{i\})$
Relabel A, B, C so that B is "top-left" of $P_{\mathcal{I}}$, C is "bottom-right" of $P_{\mathcal{I}}$, and A is under (or along) BC.
Let $S \leftarrow \{j \in N \setminus \{i\} : \frac{\beta_j}{\alpha_j} \leq \frac{\beta_i}{\alpha_i}\}$ ▷ L_j steeper than L_i
Let $G \leftarrow \{j \in N \setminus \{i\} : \frac{\beta_j}{\alpha_j} > \frac{\beta_i}{\alpha_i}\}$ ▷ L_j gentler than L_i
if $n_B \geq |S|$ **then** ▷ Case 2a
  Assign each agent in S the bundle B
  $\mathcal{I}' \leftarrow (a - |S| \cdot x(B), b - |S| \cdot y(B), N \setminus S)$
  return $\textsc{Allocate}(\mathcal{I}', D)$
if $n_C \geq |G|$ **then** ▷ Case 2b
  Assign each agent in G the bundle C
  $\mathcal{I}' \leftarrow (a - |G| \cdot x(C), b - |G| \cdot y(C), N \setminus G)$
  return $\textsc{Allocate}(\mathcal{I}', D)$
if A lies "bottom-left" of $P_{\mathcal{I}}$ **then** ▷ Case 2c
  Assign n_B agents in S the bundle B
  Assign n_C agents in G the bundle C
  Assign the remaining agents bundle A
  return
if A lies "top-left" of $P_{\mathcal{I}}$ **then** ▷ Case 2d
  Assign n_B agents in S the bundle B
  Assign $|S| - n_B$ remaining agents in S the bundle A
  $\mathcal{I}' \leftarrow (a - n_B \cdot x(B) - (|S| - n_B) \cdot x(A), b - n_B \cdot y(B) - (|S| - n_B) \cdot y(A), N \setminus S)$
  return $\textsc{Allocate}(\mathcal{I}', D)$
if A lies "bottom-right" of $P_{\mathcal{I}}$ **then** ▷ Case 2e
  Assign n_C agents in G the bundle C
  Assign $|G| - n_C$ remaining agents in G the bundle A
  $\mathcal{I}' \leftarrow (a - n_C \cdot x(C) - (|G| - n_C) \cdot x(A), b - n_C \cdot y(C) - (|G| - n_C) \cdot y(A), N \setminus G)$
  return $\textsc{Allocate}(\mathcal{I}', D)$

$(0,0), (x(P_{\mathcal{I}}), 0), P, (0, y(P_{\mathcal{I}})) \in R_j \cap \mathbb{Z}^2$ for all $j \in N$. Thus by convexity $P_{\mathcal{I}'} \in \cap_{j \in N \setminus \{i\}} IH_j$ and we can induct.

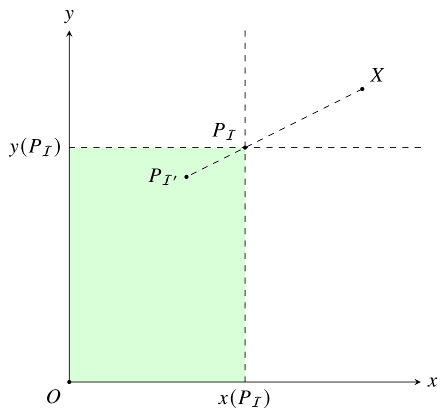


Figure 3: (Case 1). Upon assigning bundle X, the feasibility point moves from $P_{\mathcal{I}}$ to $P_{\mathcal{I}'}$.

Case 2: None of the vertices A, B, C of $\triangle_i(\mathcal{I})$ lie "top-right" of $P_{\mathcal{I}}$.

Now, observe that at least one of A, B, C must have y-coordinate which is greater than or equal to $y(P_{\mathcal{I}})$ (since the weighted centroid of A, B, C is $P_{\mathcal{I}}$). WLOG, let this be B. If we are not in Case 1, then we must have $x(B) < x(P_{\mathcal{I}})$, i.e. B is "top-left" of $P_{\mathcal{I}}$. Similarly, we may assume that C lies "bottom-right" of $P_{\mathcal{I}}$ (i.e $y(C) \leq y(P_{\mathcal{I}})$ and $x(C) > x(P_{\mathcal{I}})$), and WLOG that A lies under BC (so A, B, C are in clockwise order).

Next, define the set $S := \{j \in N \setminus \{i\} : \frac{\beta_j}{\alpha_j} \leq \frac{\beta_i}{\alpha_i}\}$ to be the set of agents $j \in N$ such that the slope of the line L_j is steeper than that of L_i. Analogously, we define $G := \{j \in N \setminus \{i\} : \frac{\beta_j}{\alpha_j} > \frac{\beta_i}{\alpha_i}\}$ to be the set of agents $j \in N$ such that the slope of the line L_j is gentler than that of L_i.

Let I_j for $j \in N$ denote the intersection point of lines L_i and L_j. For $j \in N$, let $R_j^{\text{left}} := \{(x, y) \in R_j : x \leq x(P_{\mathcal{I}})\}$ and $R_j^{\text{below}} := \{(x, y) \in R_j : y \leq y(P_{\mathcal{I}})\}$. We can make the following observation:

Observation 2. *For all $j \in S, R_i^{left} \subseteq R_j^{left}$, and for all $k \in G, R_i^{below} \subseteq R_k^{below}$ (see Fig 4).*

Proof. Let $j \in S$. We show $R_i^{\text{left}} \subseteq R_j^{\text{left}}$. Suppose $x(I_j) < x(P_{\mathcal{I}})$. Then, since L_j is steeper than L_i, we must have $v_j < v_i$ and $h_j < h_i$, but this contradicts the minimality of i under $\preceq$. Thus $x(I_j) \geq x(P_{\mathcal{I}})$. Then suppose some point $L \in R_i^{\text{left}}$. Since L lies under L_i, L_j is steeper than L_i, and $x(L) \leq x(P_{\mathcal{I}}) \leq x(I_j)$, then $L \in R_j^{\text{left}}$. The argument for $R_i^{\text{below}} \subseteq R_k^{\text{below}}$ for any $k \in G$ is symmetric. □

Note that this may not necessarily imply that when $x \leq x(P), IH_i \subseteq IH_j$, or when $y \leq y(P), IH_i \subseteq IH_k$.

Now, we use the above fact to handle the remaining cases.

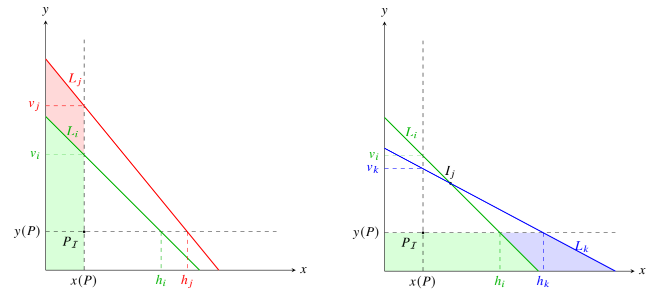


Figure 4: Illustration of Obs 2. $R_i^{\text{left}} \subseteq R_j^{\text{left}}$ for $j \in S$ (left) and $R_i^{\text{below}} \subseteq R_k^{\text{below}}$ for $k \in G$.

Case 2a: $n_B \geq |S|$

See Fig 5. Algorithm 1 assigns each agent in S the bundle B. By Observation 2, these are D_j-feasible for all $j \in S$. Let the remaining instance be $\mathcal{I}' = (a - |S| \cdot x(B), b - |S| \cdot y(B), N \setminus S)$ and note that $N \setminus S = G \cup \{i\}$. The new feasibility point $P_{\mathcal{I}'}$ must lie along the segment $\overline{P_{\mathcal{I}} Z}$ where $Z = BP_{\mathcal{I}} \cap AC$ and "bottom-right" of $P_{\mathcal{I}}$. Now since $P_{\mathcal{I}'}$ is still a weighted centroid of $A, B, C \in IH_i(D_i)$, then $P_{\mathcal{I}'} \in IH_i(D_i)$.

Fix $j \in G$. By assumption, since $y(C) \leq y(P_{\mathcal{I}})$, by Observation 2, we have $C \in R_i^{\text{below}} \subseteq R_j^{\text{below}} \subseteq R_j$. Since $C \in \mathbb{Z}^2$, then $C \in IH_j$.

Suppose $y(A) \leq y(P_{\mathcal{I}})$. Then by the same argument $A \in IH_j$. Hence since Z lies along $\overline{AC}$, by convexity, $Z \in IH_j$.

Otherwise, suppose $y(A) > y(P_{\mathcal{I}})$. Then observe that Z lies within the polygon defined by the points $(x(P_{\mathcal{I}}), 0), P_{\mathcal{I}}, C, (x(C), 0)$. Since $C \in IH_j$, then $(x(C), 0) \in IH_j$. Also, by assumption, $P_{\mathcal{I}} \in IH_j$ and hence $(x(P_{\mathcal{I}}), 0) \in IH_j$. Thus by convexity $Z \in IH_j$.

In either case, we have $Z \in \cap_{j \in N \setminus S} IH_j$, hence $P_{\mathcal{I}'} \in \cap_{j \in N \setminus S} IH_j$ by the fact that $Z, P_{\mathcal{I}} \in \cap_{j \in N \setminus S} IH_j$ and the convexity of $\cap_{j \in N \setminus S} IH_j$, and we may induct.

Case 2b: $n_C \geq |G|$

Algorithm 1 handles this symmetrically to Case 2a by assigning each agent in G the bundle C; the proof follows by exchanging the axes.

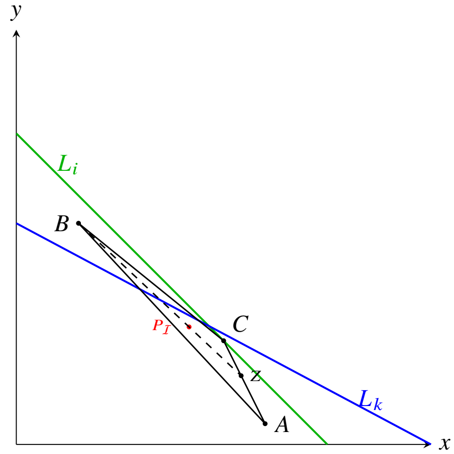
Case 2c: $n_B < |S|, n_C < |G|$ and A lies "bottom-left" of $P_{\mathcal{I}}$

See Fig 6. Algorithm 1 assigns all n_B bundles B to agents in S, and all n_C bundles to agents in G. By Observation 2, these are feasible assignments.

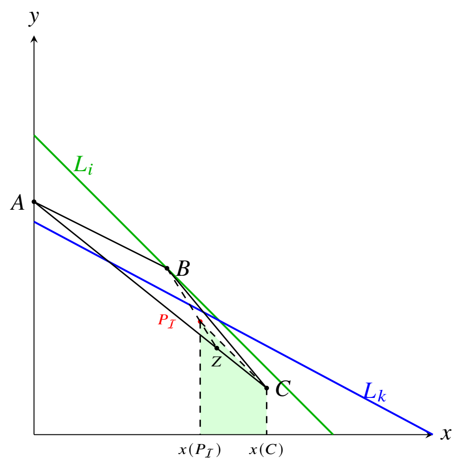
Now since A is "bottom-left" of $P_{\mathcal{I}}$, and $P_{\mathcal{I}}$ lies under $L_j(D_j)$ for all $j \in N$, i.e. A is D_j-feasible for all agents $j \in N$. Hence we may simply assign A to any remaining agent, and since $n_A \cdot A + n_B \cdot B + n_C \cdot C = (a, b)$ (by definition), this is a complete allocation and a D-feasible one.

Case 2d: $n_B < |S|, n_C < |G|$ and A lies "top-left" of $P_{\mathcal{I}}$

See Fig 7. Algorithm 1 assigns all n_B bundles B to agents in S, and $|S| - n_B$ copies of A to the remaining agents in S. Note that there are enough bundles since $n_A + n_B + n_C = |S| + |G| + 1$ and $n_C < |G|$ implies $n_A + n_B > |S|$. By Observation 2, these are D_j-feasible for all $j \in S$. Let the remaining instance be



(a) $y(A) \leq y(P_\mathcal{I})$. Red point is $P_\mathcal{I}$. Z lies on $\overline{AC}$ and $P_{\mathcal{I}'}$ lies on $\overline{P_\mathcal{I} Z}$.



(b) $y(A) > y(P_\mathcal{I})$. Red point is $P_\mathcal{I}$. $P_{\mathcal{I}'}$ lies within region bounded by $(x(P_\mathcal{I}), 0), P_\mathcal{I}, C, (x(C), 0)$.)

Figure 5: Illustration of Case 2a.

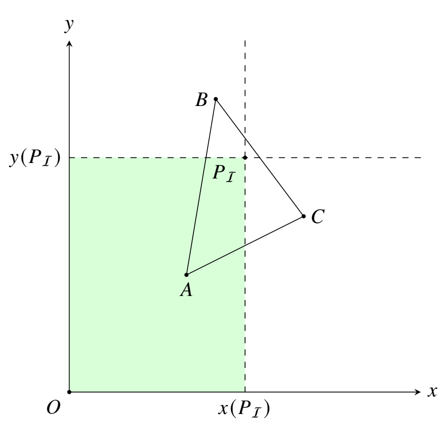


Figure 6: (Case 2c) A lies "bottom-left" of $P_\mathcal{I}$ hence is D_i-feasible for all $i \in N$.

$\mathcal{I}' = (a - n_B \cdot x(B) - (|S| - n_B) \cdot x(A), b - n_B \cdot y(B) - (|S| - n_B) \cdot y(A), N \setminus S)$ and note that $N \setminus S = G \cup \{i\}$. Let $Q := \frac{n_A \cdot A + n_C \cdot C}{n_A + n_C}$ be the weighted centroid of the remaining points after removing bundles B. This must lie along the segment $\overline{AC}$, and is "bottom-right" of $P_\mathcal{I}$. Now upon removing $|S| - n_B$ copies of A, the new feasibility point $P_{\mathcal{I}'}$ must lie along the segment $\overline{QC}$, and is still "bottom-right" of $P_\mathcal{I}$. Thus, like in Case 2a, this is within the polygon defined by the points $(x(P_\mathcal{I}), 0), P_\mathcal{I}, C, (x(C), 0)$. Then by the same argument $P_{\mathcal{I}'} \in \cap_{j \in N \setminus S} IH_j(D_j)$ and we may induct.

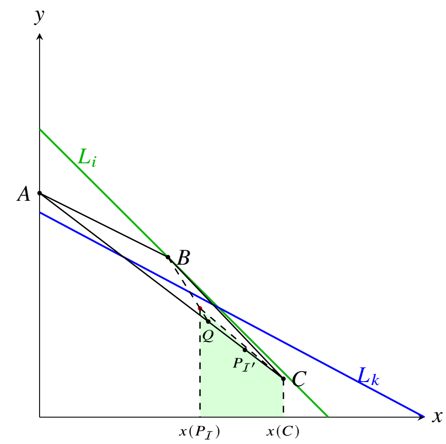


Figure 7: (Case 2d)

Case 2e: $n_B < |S|, n_C < |G|$ and A lies "bottom-right" of $P_\mathcal{I}$

Algorithm 1 handles this symmetrically to Case 2d by assigning all n_C bundles C to agents in G, and $|G| - n_C$ copies of A to the remaining agents in G. The proof follows by exchanging the axes.

□

Time complexity. By the results of McCormick et al. [1997], the values $(\text{MMS}_j)_{j \in N}$ may be computed in polynomial time via Lemma 3 and binary search. In each iteration, we may find the minimal agent $i \in N$ under $\preceq$ in $O(n)$ time. Computing $\triangle_i(\mathcal{I}, \text{MMS}_i)$ also takes $O(\log^2 \text{MMS}_i)$ time by Lemma 2, and the number of agents to consider after each iteration decreases by at least one. Thus Algorithm 1 runs in polynomial time.

4 Extension to goods and mixed items

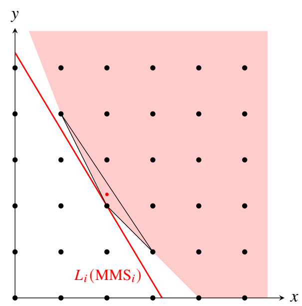
In this section, we briefly note that the procedure in Section 3 can also be modified to show the existence of MMS allocations when the two types of items are both goods, or when there is one type of item which is an *objective good* (positive utility to all agents), and another type of item which is an *objective chore* (negative utility to all agents). We handle each case separately.

Here we go back to using the regular agent valuation functions $v_i : M \to \mathbb{R}$. An item o is a good (resp. chore) for agent i if $v_i(o) > 0$ ($v_i(o) < 0$). Then recall in this setting the maximin share of an agent is defined by

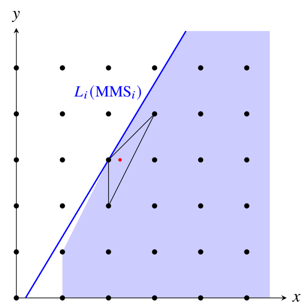
$$\mathrm{MMS}_i(M) := \max_{(X_i)_{i\in N}\in\Pi} \min_{j\in N} v_i(X_j),$$

so an MMS allocation is one where each agent receives *at least* their MMS value. We re-define t-feasibility (and analogously D-feasibility) to represent if the value of a bundle B for some agent *exceeds* some threshold t.

Observe that, in an fair-division instance with goods, the set of MMS_i-feasible bundles for agent i, $R_i(\mathrm{MMS}_i)$ now refers to the nonnegative region lying *above* the line $\alpha_i x+\beta_i y = \mathrm{MMS}_i$ (where $\alpha_i, \beta_i > 0$). Similarly, when there is one type of *objective good* and one type of *objective chore*, WLOG let Type A refer to the goods and Type B refer to the chores. Then $R_i(\mathrm{MMS}_i)$ refers to the nonnegative region lying *below* the line $\alpha_i x+\beta_i y = \mathrm{MMS}_i$ where $\alpha_i > 0$ and $\beta_i < 0$. One can show, using essentially the same proof, that in these settings Lemma 1 still holds.



(a) Goods instance. $L_i(\mathrm{MMS}_i)$ is $5x + 3y = 16$. $P_{\mathcal{I}} = (\frac{9}{4}, 2) = \frac{1}{4}((2 \cdot (2,2) + (1,4) + (3,1))$. Shaded region is IH_i.



(b) Mixed instance. $L_i(\mathrm{MMS}_i)$ is $5x - 3y = 1$. $P_{\mathcal{I}} = (\frac{9}{4}, 3) = \frac{1}{4}(2 \cdot (2,3) + (3,4) + (2,2))$. Shaded region is IH_i.

Although the regions $R_i(t)$ may now be unbounded, it is still possible to find a unimodular triangle with vertices in $R_i(t) \cap \mathbb{Z}^2$ containing $P_{\mathcal{I}}$ in polynomial time, i.e. Lemma 2 can be extended (Theorem 6.1 in [McCormick *et al.*, 1997]).

Finally, we may also define $\preceq$ analogously in each case. Recall that v_i is the value of y where $\alpha_i \cdot x(P_{\mathcal{I}}) + \beta_i \cdot y = D_i$, and h_i is the value of x where $\alpha_i x + \beta_i \cdot y(P_{\mathcal{I}}) = D_i$. For an instance of two types of goods, we say that

$$i \preceq j \iff v_i \geq v_j \text{ and } h_i \geq h_j.$$

For a mixed instance, we instead say that

$$i \preceq j \iff v_i \leq v_j \text{ and } h_i \geq h_j.$$

We omit formal descriptions of the adapted procedures; however, following the above definitions, extending Algorithm 1 to these cases essentially follow the same case analyses. Hence we have the following results.

Theorem 2. *Given an instance of fair allocation with only two types of goods, there exists an MMS allocation which can be computed in polynomial time.*

Theorem 3. *Given an instance of fair allocation where there are only two types of items; one of which being objective goods, and the other being objective chores, there exists an MMS allocation which can be computed in polynomial time.*

5 Discussion

We show the existence (and polynomial-time computation) of MMS allocations for instances with two types of chores, two types of goods, or one type of objective good and one type of objective chore, by leveraging key lemmas from prior work in job scheduling for two types of jobs.

However, our technique does not extend easily when the number of types of items is > 2, as the feasibility point $P_{\mathcal{I}}$ may not necessarily lie within some unimodular lattice simplex under the hyperplane L_i [McCormick *et al.*, 1997]. This is because Pick's theorem does not extend to dimensions $d > 2$. One possible future direction could be to consider "two-type" valuation functions more general than additive, e.g. $v_i(S) := f(x, y)$, where x and y are the number of Type A and Type B chores in S respectively.

References

[Amanatidis *et al.*, 2023] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A. Voudouris, and Xiaowei Wu. Fair division of indivisible goods: Recent progress and open questions. *Artif. Intell.*, 322(C), September 2023.

[Aziz *et al.*, 2021] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. Fair allocation of indivisible goods and chores. *Autonomous Agents and Multi-Agent Systems*, 36(1):3, November 2021.

[Aziz *et al.*, 2023] Haris Aziz, Jeremy Lindsay, Angus Ritossa, and Mashbat Suzuki. Fair Allocation of Two Types of Chores. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, pages 143–151, May 2023.

[Aziz, 2020] Haris Aziz. Developments in multi-agent fair allocation. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 13563–13568, April 2020.

[Barman and Verma, 2021] Siddharth Barman and Paritosh Verma. Existence and Computation of Maximin Fair Allocations Under Matroid-Rank Valuations. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '21, pages 169–177, May 2021.

[Barman *et al.*, 2023] Siddharth Barman, Vishnu Narayan, and Paritosh Verma. Fair Chore Division under Binary Supermodular Costs. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, pages 2863–2865, May 2023.

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

[Feige *et al.*, 2021] Uriel Feige, Ariel Sapir, and Laliv Tauber. A Tight Negative Example for MMS Fair Allocations. In *Proceedings of the 17th International Conference on Web and Internet Economics (WINE)*, pages 355–372, December 2021.

[Feige, 2022] Uriel Feige. Maximin fair allocations with two item values. *Unpublished Manuscript*, 2022.

[Garg *et al.*, 2025] Jugal Garg, Xin Huang, and Erel Segal-Halevi. Improved maximin share approximations for chores by bin packing. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence*, pages 13881–13888, August 2025.

[Gates *et al.*, 2020] Victoria Gates, Thomas L. Griffiths, and Anca D. Dragan. How to be helpful to multiple people at once. *Cognitive Science*, 44(6):e12841, 2020.

[Gorantla *et al.*, 2023] Pranay Gorantla, Kunal Marwaha, and Santhoshini Velusamy. Fair allocation of a multiset of indivisible items. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 304–331, January 2023.

[Huang and Segal-Halevi, 2023] Xin Huang and Erel Segal-Halevi. A Reduction from Chores Allocation to Job Scheduling. In *Proceedings of the 24th ACM Conference on Economics and Computation*, EC '23, page 908, July 2023.

[Huang and Zhou, 2025] Xin Huang and Shengwei Zhou. An fptas for 7/9-approximation to maximin share allocations, 2025.

[Kurokawa *et al.*, 2018] David Kurokawa, Ariel D. Procaccia, and Junxing Wang. Fair Enough: Guaranteeing Approximate Maximin Shares. *J. ACM*, 65(2):8:1–8:27, February 2018.

[Liu *et al.*, 2024] Shengxin Liu, Xinhang Lu, Mashbat Suzuki, and Toby Walsh. Mixed fair division: A survey. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(20):22641–22649, March 2024.

[McCormick *et al.*, 1997] S. Thomas McCormick, Scott R. Smallwood, and Frits C. R. Spieksma. Polynomial algorithms for multiprocessor scheduling with a small number of job lengths. In *Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms*, SODA '97, pages 509–517, January 1997.

[Viswanathan and Zick, 2025] Vignesh Viswanathan and Yair Zick. A General Framework for Fair Allocation under Matroid Rank Valuations. *ACM Trans. Econ. Comput.*, 13(3):14:1–14:32, September 2025.