Keeping the Harmony Between Neighbors: Local Fairness in Graph Fair Division

Halvard Hummel
Norwegian University of Science and Technology
Trondheim, Norway
halvard.hummel@ntnu.no

Ayumi Igarashi
University of Tokyo
Tokyo, Japan
igarashi@mist.i.u-tokyo.ac.jp

ABSTRACT

We study the problem of allocating indivisible resources under the connectivity constraints of a graph G. This model, initially introduced by Bouveret et al. (published in IJCAI, 2017), effectively encompasses a diverse array of scenarios characterized by spatial or temporal limitations, including the division of land plots and the allocation of time plots. In this paper, we introduce a novel fairness concept that integrates local comparisons within the social network formed by a connected allocation of the item graph. Our particular focus is to achieve pairwise-maximin fair share (PMMS) among the "neighbors" within this network. For any underlying graph structure, we show that a connected allocation that maximizes Nash welfare guarantees a $(1/2)$-PMMS fairness. Moreover, for two agents, we establish that a $(3/4)$-PMMS allocation can be efficiently computed. Additionally, we demonstrate that for three agents and the items aligned on a path, a PMMS allocation is always attainable and can be computed in polynomial time. Lastly, when agents have identical additive utilities, we present a pseudo-polynomial-time algorithm for a $(3/4)$-PMMS allocation, irrespective of the underlying graph G. Furthermore, we provide a polynomial-time algorithm for obtaining a PMMS allocation when G is a tree.

KEYWORDS

Fair division; Pairwise-maximin fair share; Graph

ACM Reference Format:
Halvard Hummel and Ayumi Igarashi. 2024. Keeping the Harmony Between Neighbors: Local Fairness in Graph Fair Division. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

1 INTRODUCTION

Consider the distribution of offices among research groups. This task is not simple due to the inherent differences in size, quality, and location of the offices. Moreover, individuals may possess diverse preferences for these factors. For instance, certain individuals may prioritize office size, as they require sufficient space to conduct experiments, while others may emphasize proximity to student canteens. In light of these considerations, how can one effectively address individuals' needs and ensure a "fair" distribution of offices?

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

This question has been explored in the extensive literature on fair division [28]. The existing literature has traditionally focused on two primary fairness notions: envy-freeness and proportionality. Evny-freeness requires each agent to prefer her own bundle over the bundle of anyone else. Proportionality ensures that each agent receives a fair share, comprising at least $1/n$ of the total utility of the resource. However, achieving either of these notions becomes challenging when the resources are indivisible. For instance, when distributing a single office among two agents, we can satisfy neither of the fairness requirements. To address this limitation, recent research has explored alternative fairness measures that relax these requirements. Examples include envy-freeness up to one item (EF1) and maximin fair share (MMS) [14].

In this paper, we investigate a model that focuses on the *connected* allocation of indivisible resources arranged in a graph structure. This model, as proposed by Bouveret et al. [10], captures various resource allocation scenarios that involve spatial or temporal constraints. Using the example of office allocation, it is not desirable for members of a research group to be assigned offices that are widely dispersed. The vertices of the graph can represent not only offices but also land plots, time slots, and other similar resources.

While envy-freeness and proportionality are natural fairness concepts within the graph fair division model, it may not be realistic to assume that individuals make global comparisons across the entire resource. According to the theory of social comparison, people often engage in local comparisons, where they assess their situation in relation to their peers, neighbors, or family members [17, 31, 32].

In the context of fair division, several papers have explored this phenomenon by assuming that agents are part of a social network [1, 6, 7, 13, 20]. Much of this research focuses on local fairness requirements, where agents compare their allocation only with that of their neighbors within the underlying social network.

To integrate these two elements, fair division *of* and *over* graphs, we present a novel model that incorporates local comparison into the graph fair division setting. More precisely, we examine the social network formed by a connected allocation of the item graph and investigate local fairness among the "neighbors" in the network. In this context, two agents are considered neighbors if they are allocated to vertices that are adjacent to each other. In practical scenarios, this could correspond to neighboring countries, employees assigned to consecutive shifts, or research groups with adjacent offices. Building upon recent research on approximate fairness [14, 15], we focus on the local variant of pairwise maximin fair share (PMMS). We aim at addressing the following question:

> *What local fairness guarantees can be achieved under various types of graphs? Can we simultaneously attain local and global fairness?*

Table 1: Summary of our results for additive utilities. The non-existence † holds due to a counterexample for MMS existence on a graph [10, 25]. The existence ⊕ follows from the cut-and-choose argument among two agents with additive utilities. The NP-hardness ⊗ is due to the NP-hardness of Partition.

Properties		*n agents (any graph)*	*2 agents (any graph)*	*3 agents (path)*	*Identical agents*
MMS & PMMS	Existence	No†	Yes⊕	Open	Yes (Th. 6.1)
	Computation	NP-hard⊗	NP-hard⊗	Open	Poly for trees (Cor. 6.8)
α-PMMS	Existence	Yes for $\alpha = 1/2$ (Th. 3.1)	Yes for $\alpha = 1$ ⊕	Yes for $\alpha = 1$ (Th. 5.1)	Yes for $\alpha = 1$ (Th. 6.1)
	Computation	Open for constant $\alpha \in (0, 1)$	Poly for $\alpha = 3/4$ (Th. 4.1)	Poly (Th. 5.1)	Pseudo-poly for $\alpha = 3/4$ (Th. 6.3)

1.1 Our contributions

In Section 2, we present the formal model and introduce our local fairness concept of pairwise maximin fair share (PMMS), originally introduced by Caragiannis et al. [15]. Intuitively, it requires that each agent receives a fair share, determined by redistributing her bundle and the bundle of her neighbor via the the cut-and-choose algorithm. More precisely, the PMMS of agent i with respect to her neighbor j is the maximum utility i can achieve if she were to partition the combined bundle of the pair into two and choose the worst bundle. While PMMS implies (4/7)-MMS in the standard setting of fair division [3], we show that there is no implication relation between PMMS and MMS up to any multiplicative factor in the graph-restricted setting. In the full paper, we further discuss the relationship between PMMS and other solution concepts.[1]

In Section 3, we establish the strong compatibility between the local fairness notion of PMMS and the global criterion of Pareto-optimality. We show that for n agents with additive utilities and general graphs, any MNW allocation that maximizes the Nash welfare (i.e., the product of utilities) satisfies (1/2)-PMMS. This stands in sharp contrast to an impossibility result regarding EF1 that α-EF1 with any $\alpha > 0$ does not exist even for two agents and a star graph G; see Proposition 3.3.

In Section 4, we focus on the case of two agents with additive utilities. The two-agent case serves as a foundational model of the fair division problem and has wide-ranging practical applications, including divorce settlements and inheritance division [11, 12, 23]. While computing a PMMS allocation is in general hard even for two agents, we show that (3/4)-PMMS allocation can be computed in polynomial time for any connected graph G. Our proof crucially rests on an important observation that for biconnected graphs, a (3/4)-PMMS allocation can be computed efficiently by constructing a bipolar number over the graph. By exploiting the acyclic structure of the maximal biconnected subgraphs (called blocks) of G, we show that our problem can in principle be reduced to that for some valuable block of G.[2]

In Section 5, we consider the case of three agents with additive utilities. For three agents with non-identical additive utilities on a path, we show that a PMMS connected allocation exists and can be computed in polynomial time. Note that in the standard setting of fair division, the existence of PMMS allocations remains an open question, particularly for three agents with additive utilities.

In Section 6, we focus on the case of identical additive utilities. We show that when agents have identical utility, there exists a pseudo-polynomial time algorithm that computes a (3/4)-PMMS allocation of any connected graph. Further, if we add the condition that the underlying graph is a tree, we can compute an allocation that simultaneously satisfies both PMMS and MMS in polynomial time. In fact, our final allocation satisfies a stronger property of MMS where the number of agents receiving their exact maximin fair share is minimized. See Table 1 for an overview of our results.

Related work Our work aligns with the growing literature on fair allocation of and over graphs.

Abebe et al. [1] and Bei et al. [6] initiated the study of fair allocation of divisible resources with *agents* arranged on a social network. They introduced the concepts of local envy-freeness and local proportionality, which restrict comparisons to pairs of neighboring agents. Bredereck et al. [13] investigated local fairness in the context of indivisible resource allocation, providing computational complexity results for local envy-freeness. Beynier et al. [7] and Hosseini et al. [20] examined house allocation problems over social networks where each agent can be allocated at most one item. However, these works assume that the social network is predetermined, whereas in our work, the network structure is not predefined.

In parallel, several papers have studied fair allocation of resources aligned on a graph [10, 19, 21, 22, 27, 33]. It has been shown that for well-structured classes of graphs, a connected allocation satisfying global fairness can always be achieved. For example, Bouveret et al. [10] demonstrated that for trees, a connected allocation satisfying maximin fair share (MMS) always exists. Another approximate fairness notion, envy-freeness up to one item (EF1), can be attained under connectivity constraints of a path [8, 21]. However, the existence guarantee does not hold for general graphs: an MMS connected allocation may not exist on cycles [10], and an EF1 connected allocation is not guaranteed even for a star graph with two agents having identical binary utilities [8].

Caragiannis et al. [15] introduced the concept of PMMS and identified its connection with other solution concepts in the standard setting of fair division without connectivity constraints. They showed that any PMMS allocation satisfies an approximate notion of envy-freeness, called EFX. Furthermore, they proved that any maximum Nash welfare (MNW) allocation always satisfies α-PMMS where $\alpha \approx 0.618$ is the golden ratio conjugate. Amanatidis et al. [4] improved this factor to $\frac{2}{3}$ and provided a polynomial-time algorithm for computing such an allocation. The current best multiplicative factor α for which α-PMMS is known to exist in the standard setting of fair division is 0.781 by Kurokawa [24]. Amanatidis et al. [3]

[1] The full paper is available on arXiv: https://arxiv.org/abs/2401.14825.

[2] A similar technique using blocks has been used to characterize a family of graphs for which a connected allocation among two agents satisfying EF1 (and its relaxations) is guaranteed to exist [5, 8].

discussed the relationship between approximate versions of PMMS and several fairness concepts, such as EF1 and EFX. See also [2] for a recent survey on fair division.

2 MODEL AND FAIRNESS CONCEPTS

We define a fair division problem of indivisible items where the items are placed on a graph. For a natural number $s \in \mathbb{N}$, we write $[s] = \{1, 2, \ldots, s\}$. Let $N = [n]$ be a set of *agents* and $G = (V, E)$ an *item graph*. Throughout this paper, we assume that G is a simple graph, i.e., G does not contain loops or parallel edges. Each agent i has a *utility function* $u_i : 2^V \to \mathbb{R}_+$. For simplicity, the utility of a single vertex $v \in V$, $u_i(\{v\})$, is also denoted by $u_i(v)$. The elements in V are referred to as *items* (or *vertices*). Each subset $X \subseteq V$ is referred to as a *bundle* of items. A bundle $X \subseteq V$ is *connected* if it is a connected subgraph of G. An *allocation* $A = (A_1, A_2, \ldots, A_n)$ is a partition of the items into disjoint bundles of items, i.e., $\bigcup_{i \in N} A_i = V$ and $A_i \cap A_j = \emptyset$ for every pair of distinct agents $i, j \in N$. We say that an allocation A is *connected* if for every $i \in N$, A_i is connected in G. A utility function u_i is *monotone* if $u_i(X) \leq u_i(Y)$ for every $X \subseteq Y \subseteq V$. It is *additive* if $u_i(X) = \sum_{v \in X} u_i(v)$ holds for every $X \subseteq V$ and $i \in N$. An additive utility u_i is *binary additive* if $u_i(v) \in \{0, 1\}$ for every $v \in V$ and $i \in N$.

We introduce fairness notions including those based on local comparison among neighbors and those based on global comparison among agents. Given an allocation A, we say that $i, j \in N$ are *adjacent* under A if there is an edge $\{v_1, v_2\} \in E$ such that $v_1 \in A_i$ and $v_2 \in A_j$; we call j a *neighbor* of i. For every connected bundle $X \subseteq V$ and natural number k, we denote by $\Pi_k(X)$ the set of partitions of X into k connected subgraphs of a graph $G[X]$, where $G[X]$ is the subgraph of G induced by X. For agent i, a connected bundle X, and natural number k, we define the *k-maximin share* of agent i with respect to X as $\mu_i^k(X) = \max\{\, \min_{j \in [k]} u_i(A_j) \mid A \in \Pi_k(X)\,\}$. For $k = 2$ and $k = n$, we write $\mu_i^2(X)$ and $\mu_i^n(X)$ as $\mathrm{PMMS}_i(X)$ and $\mathrm{MMS}_i(X)$, respectively. For agents with identical utility function u, we simply write $\mathrm{PMMS}(X) = \mathrm{PMMS}_i(X)$ and $\mathrm{MMS}(X) = \mathrm{MMS}_i(X)$ for each agent i.

Definition 2.1 (Pairwise MMS). An allocation A is called α-PMMS if for every pair of agents $i, j \in N$ such that $A_i = \emptyset$ or i and j are adjacent under A, $u_i(A_i) \geq \alpha \cdot \mathrm{PMMS}_i(A_i \cup A_j)$. For $\alpha = 1$, we refer to the corresponding allocations as PMMS allocations.

In order to avoid a trivial allocation (allocating all items to one agent) from satisfying PMMS, we allow agents with an empty bundle to compare their bundle with every other agent's bundle under the definition above. Note that under additive utilities we have $\frac{1}{2}u_i(X) \geq \mathrm{PMMS}_i(X)$, since in any partition of a set X into two sets, one bundle always has a utility of at most a half of the total utility with respect u_i. We say that a partition $A = (A_1, A_2, \ldots, A_n)$ of G is a *PMMS partition* of agent i if A_j is connected for each $j \in [n]$, and it is PMMS for agent i no matter which bundle i receives. An allocation A is called α-MMS if for every agent $i \in N$, $u_i(A_i) \geq \alpha \cdot \mathrm{MMS}_i(V)$. For $\alpha = 1$, we refer to the corresponding allocations as MMS allocations.

Note that there is no implication relation between PMMS and MMS even for a path and agents with identical additive utilities.

Proposition 2.2. *Even for a path and three agents with additive identical utilities, neither PMMS nor MMS implies the other up to any multiplicative factor* $\alpha \in (0, 1]$.

Proof. Consider any $\alpha \in (0, 1]$. Choose β such that $\alpha > \frac{1}{\beta}$. Note that $\beta > 1$. To show that MMS may not satisfy α-PMMS, consider an instance of three agents with utility function u and four vertices on a path. Each agent has utility 1, β, β, and 1 for each of the vertices, starting from the leftmost. Consider an allocation A that allocates the leftmost item to agent 1, the rightmost item to agent 2, and the remaining items to agent 3. Here, the maximin fair share for each agent is 1 and thus this allocation satisfies MMS. However, A is not α-PMMS. Indeed, we have $\frac{1}{\alpha} = \frac{1}{\alpha}u(A_1) < \mathrm{PMMS}(A_1 \cup A_2) = \beta$.

To show that PMMS may not satisfy α-MMS, consider an instance of three agents with utility function u and four vertices on a path. Each agent has utility 1, β, β, and β for each of the vertices, starting from the leftmost. Consider an allocation A that allocates the leftmost item to agent 1, the second leftmost item to agent 2, and the remaining items to agent 3. Here, the allocation satsfies PMMS. However, A is not α-MMS. Indeed, we have $\frac{1}{\alpha} = \frac{1}{\alpha}u(A_1) < \mathrm{MMS}(V) = \beta$. □

An allocation A is α-EF1 if for any pair of agents $i, j \in N$ with $A_j \neq \emptyset$, there exists an item $v \in A_j$ such that $u_i(A_i) \geq \alpha \cdot u_i(A_j \setminus \{v\})$. For $\alpha = 1$, we refer to the corresponding allocations as EF1 allocations. Given an allocation A, another allocation A' is a *Pareto-improvement* of A if $u_i(A'_i) \geq u_i(A_i)$ for every $i \in N$ and $u_j(A'_j) > u_j(A_j)$ for some $j \in N$. We say that a connected allocation A is *Pareto-optimal* if there is no connected allocation that is a Pareto-improvement of A.

We say that a connected allocation A is a *maximum Nash welfare (MNW)* allocation if it maximizes the number of agents receiving positive utility and, subject to that, maximizes the product of the positive utilities, i.e., $\prod_{i \in N : u_i(A_i) > 0} u_i(A_i)$, over all connected allocations. For an allocation A, let $\mathbf{u}(A)$ be the vector obtained from rearranging the elements of the vector $(u_1(A_1), u_2(A_2), \ldots, u_n(A_n))$ in increasing order. Given two allocations A and A', an allocation A is a *leximin improvement* of A' if there exists $k \in [n]$ such that the first $k - 1$ elements of $\mathbf{u}(A)$ and $\mathbf{u}(A')$ are the same, but the k-th element of $\mathbf{u}(A)$ is greater than that of $\mathbf{u}(A')$. A connected allocation A is called *leximin* if there is no connected allocation that is a leximin improvement of A.

3 THE CASE OF n AGENTS

In this section, we consider the case of n agents. We start by showing that any MNW connected allocation satisfies $(1/2)$-PMMS.

Theorem 3.1. *For a connected graph G and n agents with additive utilities, any MNW connected allocation satisfies* $(1/2)$*-PMMS.*

Proof. For a connected graph G, let A^* be any MNW connected allocation. Then take any pair of agents i and j such that $A^*_i = \emptyset$, or i and j are adjacent under A^*. Let $M_{ij} = A^*_i \cup A^*_j$. The allocation $A^*_{ij} = (A^*_i, A^*_j)$ is an MNW allocation of $G[M_{ij}]$ to i and j. We wish to show that $u_i(A^*_i) \geq \mathrm{PMMS}_i(M_{ij})/2$ and $u_j(A^*_j) \geq \mathrm{PMMS}_i(M_{ij})/2$. If $u_i(A^*_i) = 0$ or $u_j(A^*_j) = 0$, then j or i must receive a bundle with utility $u_j(M_{ij})$ and $u_i(M_{ij})$, respectively. Thus, in either case, both agents receive a bundle with utility at least their PMMS.

Assume now that $u_i(A_i^*) > 0$ and $u_j(A_j^*) > 0$. Let (A_1, A_2) be a PMMS partition of M_{ij} for agent i. Then at least one of the bundles has utility at least $u_j(M_{ij})/2$ according to u_j. Let $A = (A_i, A_j)$ be the allocation in which j receives this bundle and i the other bundle. Then $u_i(A_i) \geq \text{PMMS}_i(M_{ij})$. Moreover, it holds that $u_i(A_i^*) \cdot u_j(A_j^*) \geq u_i(A_i) \cdot u_j(A_j)$. Substituting $u_j(A_j^*)$ by an upper bound, and $u_i(A_i)$ and $u_j(A_j)$ by lower bounds, $u_i(A_i^*) \cdot u_j(M_{ij}) \geq \text{PMMS}_i(M_{ij}) \cdot u_j(M_{ij})/2$, which implies $u_i(A_i^*) \geq \text{PMMS}_i(M_{ij})/2$. Thus, agent i receives in A^* a bundle with utility at least a half of her PMMS. By exchanging i and j in the argument, j must also receives in A^* a bundle with utility at least a half of her PMMS. □

Corollary 3.2. *For a connected graph G and n agents with additive utilities, a Pareto-optimal and $(1/2)$-PMMS connected allocation exists.*

The one (1/2)-PMMS guarantee in Theorem 3.1 is unfortunately the best that we can hope for using MNW allocations. Even if instances are restricted to two agents on a path with binary additive utilities, there exist instances with MNW allocations that provide one of the agents with exactly half their PMMS. Further, a leximin allocation does not satisfy any approximation of PMMS, even when agents have binary additive utilities on paths. See Propositions 3.3 and 3.4 in the full paper.

Below, we show that unlike PMMS, we cannot guarantee an α-EF1 connected allocation, even among two agents for any $\alpha > 0$.

Proposition 3.3. *For any $\alpha \in (0, 1]$, there exists an instance of a star and two agents with identical binary additive utilities such that no α-EF1 connected allocation exists.*

Note that Theorem 3.1 does not provide us an efficient algorithm to compute a $(1/2)$-PMMS connected allocation. In fact, finding an MNW allocation is NP-hard even when G is a tree and agents have binary additive utilities [22]. In the next sections, we identify several cases in which an approximate/exact PMMS connected allocation can be efficiently computed.

4 THE CASE OF TWO AGENTS

For any connected graph G and two agents with additive utilities, a PMMS connected allocation can always be constructed by the following cut-and-choose procedure: let one agent compute his or her PMMS partition and another agent choose a preferred bundle, leaving the remainder for the first agent.[3] However, computing such a partition can easily be shown to be NP-hard by a simple reduction from PARTITION. This leads us to the following question: can we obtain good approximation guarantees with respect to PMMS in polynomial time? We answer this question affirmatively for two agents, by showing that a $(3/4)$-PMMS connected allocation among two agents can always be computed in polynomial time for any additive utility functions. Note that for two agents, PMMS and MMS coincide with each other, and thus the $(3/4)$-PMMS guarantee is equivalent to $(3/4)$-MMS.

Theorem 4.1. *For a connected graph G and two agents with additive utility functions, a $(3/4)$-PMMS (equivalently, $(3/4)$-MMS) connected allocation can be found in polynomial time.*

[3]Note that this method does not work for non-additive utilities. Indeed, even for two agents with submodular utilities on a cycle, there is an instance for which a PMMS connected allocation does not exist. See Proposition A.1 in the full paper.

Algorithm 1 Algorithm for constructing a $(3/4)$-PMMS connected allocation for two agents with identical additive utilities u when $G = (V, E)$ is biconnected

1: Choose vertices v^*, w^* with highest, $u(v^*)$, and second highest, $u(w^*)$, utility in G. Set $Y = \{v^*\}$ and $X = V \setminus Y$.
2: **if** $u(Y) < \frac{3}{8}u(V)$ **then**
3: Compute a bipolar ordering $(v_1, v_2, \dots, v_k)$ over G with $v_1 = v^*$ and $v_k = w^*$.
4: Find smallest j with $u(\{v_1, \dots, v_j\}) \geq u(\{v_{j+1}, \dots, v_k\})$.
5: **if** $u(\{v_1, \dots, v_{j-1}\}) \geq u(\{v_{j+1}, \dots, v_k\})$ **then**
6: Set $Y = \{v_1, \dots, v_{j-1}\}$ and $X = \{v_j, \dots, v_k\}$.
7: **else**
8: Set $Y = \{v_1, \dots, v_j\}$ and $X = \{v_{j+1}, \dots, v_k\}$.
9: **end if**
10: **end if**
11: **return** (X, Y)

To prove Theorem 4.1, it suffices to show that a $(3/4)$-PMMS connected allocation among two agents with identical utility can be computed in polynomial time.

Theorem 4.2. *For a connected graph G and two agents with identical additive utility function u, a $(3/4)$-PMMS (equivalently, $(3/4)$-MMS) connected allocation can be found in polynomial time.*

We first observe that whenever G is *biconnected*, a $(3/4)$-PMMS connected allocation can be computed in polynomial time. Formally, we say that a vertex v is a *cut vertex* of a connected graph G if removing v makes G disconnected. A graph G is *biconnected* if G does not have a cut vertex. A *bipolar ordering* of a graph G is an enumeration $v_1, v_2, \dots, v_k$ of G's vertices such that the subgraph induced by any initial or final segment of the enumeration is connected in G, i.e., both $\{v_1, \dots, v_j\}$ and $\{v_{j+1}, \dots, v_k\}$ are connected in G for every $j \in [k]$. It is known that G admits a bipolar ordering between any pair of vertices if G is biconnected.

Lemma 4.3 ([26]). *For a biconnected graph G and any pair v, w of vertices in G, a bipolar ordering over G that starts with v and ends with w exists and can be computed in polynomial time.*

Consider Algorithm 1. For a biconnected graph G, the algorithm either chooses a valuable item and its complement to construct a $(3/4)$-PMMS connected allocation, or computes a bipolar ordering over the graph and applies the discrete cut-and-choose algorithm.[4] Note that Line 4 of the algorithm is well-defined since $u(\{v_1, \dots, v_k\}) \geq u(\emptyset)$.

Lemma 4.4. *Let G be a biconnected graph. Suppose that there are two agents with identical additive utility function u. Then, Algorithm 1 computes in polynomial time, a $(3/4)$-PMMS connected allocation (X, Y) where*

(i) *Y consists of a single vertex and $u(Y) \geq \frac{3}{8}u(V)$,*
(ii) *$\min\{u(X), u(Y)\} \geq \frac{3}{8}u(V)$, or*
(iii) *$\min\{u(X), u(Y)\} \geq \frac{1}{2}u(V \setminus \{z^*\})$, where $z^* \in \text{argmax}\{\, u(v) \mid v \in V \setminus \{v^*, w^*\} \,\}$.*

[4]This algorithm is used to construct an EF1 connected allocation of G possessing a bipolar ordering; see Proposition 3.4 of [8].

Proof. By Lemma 4.3, Algorithm 1 clearly runs in polynomial time. To see that it returns a $(3/4)$-PMMS connected allocation, let (X, Y) be the resulting allocation of Algorithm 1.

Suppose that there is a vertex v with $u(\{v\}) \geq \frac{3}{8}u(V)$. Then, by the choice of v^* in Line 1, $u(\{v^*\}) \geq \frac{3}{8}u(V)$ and the resulting allocation (X, Y) satisfies (i). This also means that $u(\{v^*\}) \geq \frac{3}{4}(\frac{1}{2}u(V)) \geq \frac{3}{4}\text{PMMS}(V)$. Further, in any PMMS partition (X', Y'), at least one part is contained in $X = V \setminus \{v^*\}$, meaning that $u(X)$ is at least PMMS(V). Thus, (X, Y) is $(3/4)$-PMMS. Since G is biconnected, X is connected in G and (X, Y) is a connected allocation.

Suppose that $u(\{v\}) < \frac{3}{8}u(V)$ for each vertex $v \in V$. In this case, the algorithm computes a bipolar ordering in Line 3. By definition of a bipolar ordering, (X, Y) is a connected allocation. Observe that there is at most one vertex $z^* \in V \setminus \{v^*, w^*\}$ with utility $u(z^*) > \frac{1}{4}u(V)$ since $u(V \setminus \{v^*, w^*\}) < \frac{1}{2}u(V)$.

If there is no such vertex, by the fact that $\min\{u(X), u(Y)\} \geq \max\{u(X), u(Y)\} - u(v_j)$ and $u(V \setminus \{v_j\}) \geq \frac{3}{4}u(V)$, we obtain $\min\{u(X), u(Y)\} \geq \frac{1}{2}u(V\setminus\{v_j\}) \geq \frac{1}{2}(\frac{3}{4}u(V))$ and (ii) holds, which means that $\min\{u(X), u(Y)\} \geq \frac{3}{4}(\frac{1}{2}u(V)) \geq \frac{3}{4}\text{PMMS}(V)$.

Now, consider the case when there is a vertex $z^* \in V \setminus \{v^*, w^*\}$ with utility $u(z^*) > \frac{1}{4}u(V)$. Then, the vertex v_j computed in Line 4 of the algorithm has utility at most $u(z^*)$. Indeed, since $u(v^*) < \frac{3}{8}u(V) < \frac{1}{2}u(V)$, we have $u(v_1) < u(V \setminus \{v_1\})$ and thus $j > 1$. Also, since $u(w^*) < \frac{1}{2}u(V)$, we have $u(\{v_1, \ldots, v_{k-1}\}) > u(\{v_k\})$ and thus $j < k$. Thus, we obtain $\min\{u(X), u(Y)\} \geq \frac{1}{2}u(V \setminus \{v_j\}) \geq \frac{1}{2}u(V \setminus \{z^*\})$ and (iii) holds. We claim that $\text{PMMS}(V) \leq u(V) - 2u(z^*)$. Indeed, in any PMMS partition, one bundle contains at least two vertices of $\{v^*, w^*, z^*\}$, each of which has utility at least $u(z^*)$. Therefore, the other bundle has utility at most $u(V) - 2u(z^*)$. Thus,

$$\begin{aligned}\min\{u(X), u(Y)\} &\geq \frac{1}{2}u(V \setminus \{z^*\}) \\ &\geq \frac{1}{2}(u(V) - 2u(z^*)) + \frac{1}{2}u(z^*) \\ &\geq \frac{1}{2}\text{PMMS}(V) + \frac{1}{8}u(V) \\ &\geq \frac{1}{2}\text{PMMS}(V) + \frac{1}{4}\text{PMMS}(V) \\ &\geq \frac{3}{4}\text{PMMS}(V).\end{aligned}$$

We conclude that (X, Y) is a $(3/4)$-PMMS connected allocation. □

For the case when G is not necessarily biconnected, we cannot use the technique described in the proof of Lemma 4.4, since removing a single vertex of G may disconnect the graph or G may not admit a bipolar ordering (e.g., G may be a star). Nevertheless, we can exploit the acyclic structure of the so-called *block decomposition* and obtain sufficient conditions under which a PMMS connected allocation exists. Formally, a *decomposition* of a graph $G = (V, E)$ is a family $\{F_1, F_2, \ldots, F_t\}$ of edge-disjoint subgraphs of G such that $\bigcup_{i=1}^{t} E(F_i) = E$ where $E(F_i)$ is the set of edges of F_i. A *block* of G is a maximal biconnected subgraph of G.

For a connected graph G, consider a bipartite graph $B(G)$ with bipartition $(\mathcal{B}, S)$, where $\mathcal{B}$ is the set of blocks of G and S the set of cut vertices of G; a block B and a cut vertex v are adjacent in $B(G)$ if and only if v belongs to B. Since every cycle of a graph is included in some block, the graph $B(G)$ is a tree:

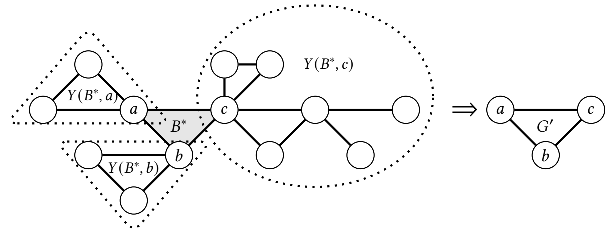


Figure 1: Merge operation applied to a block B^*.

Lemma 4.5 (Prop. 5.3 in [9]). *The set of blocks forms a decomposition of a connected graph G and the graph $B(G)$ is a tree.*

Now, observe that since $B(G)$ is a tree, removing an edge between an arbitrary cut vertex c and its adjacent block B in $B(G)$ results in two connected components X' and Y', where one part X' contains B and another part Y' contains c. This partition of the block graph induces a partition $(X(B, c), Y(B, c))$ of the original vertices in G, where $X(B, c)$ is the set of vertices in G that belong to X' except for c and $Y(B, c)$ is the set of vertices in G that belong to Y'. See Figure 1 in the full paper for an illustration.

It turns out that if no "local" improvement is possible, an allocation of form $(X(B, c), Y(B, c))$ is in fact a PMMS connected allocation with respect to identical utility function u. We defer the proof to the full paper.

Lemma 4.6. *Let G be a connected graph. Suppose that there are two agents with identical additive utility function u. Let (B, c) be a pair of a block B and a cut vertex c included in B such that $u(X(B, c)) \leq u(Y(B, c))$ and $u(X(B, c)) \geq \min\{u(X(B', c)), u(Y(B', c))\}$ for every block B' containing c. Then, the partition $(X(B, c)), Y(B, c))$ is a PMMS connected allocation.*

Now, we are ready to prove Theorem 4.2.

Proof sketch of Theorem 4.2. Let G be a connected graph and u be an identical additive utility function. If G is biconnected, by Lemma 4.4, a $(3/4)$-PMMS connected allocation can be found in polynomial time. Suppose that G is not biconnected. Let (B^*, c^*) be a pair of block and cut vertex where $(X(B^*, c^*), Y(B^*, c^*))$ maximizes $\min\{u(X(B, c)), u(Y(B, c))\}$ over all pairs of cut vertices c and its adjacent block B. If we have $u(X(B^*, c^*)) \leq u(Y(B^*, c^*))$, then $(X(B^*, c^*), Y(B^*, c^*))$ is a PMMS partition by Lemma 4.6 and by the choice of (B^*, c^*).

Thus, assume that $u(X(B^*, c^*)) \geq u(Y(B^*, c^*))$. For each cut vertex c adjacent to B^*, merge $Y(B^*, c)$ into c, namely, we replace the vertices in $Y(B^*, c)$ with a single vertex c and there is an edge between c and another vertex w in the new graph whenever w is adjacent to c in the original graph. Let G' denote the resulting graph. See Figure 1 for an illustration. It is easy to see that G' is biconnected. Moreover, we define a new additive utility function u' on G' where the utility of an agent for each vertex v of G' is equal to her utility for all the vertices of G that are merged into v. Note that for any connected allocation (X, Y) of G', it is not difficult to see that the corresponding allocation (X^*, Y^*) of the original vertices in G is connected.

Apply Algorithm 1 for G' and u'. Let (X, Y) denote the output of Algorithm 1 for G' and u'. Let us denote by X^* (resp. Y^*) the set of vertices of G merged into some vertex in X (resp. Y). As discussed before, (X^*, Y^*) is a connected allocation in G. Similar to the proof of Lemma 4.4, we can show that (X^*, Y^*) is a (3/4)-PMMS connected allocation of the original vertices. Importantly, the assumption that $u(X(B^*, c^*)) \geq u(Y(B^*, c^*))$ guarantees that even if there is a single merged vertex with high utility, its complement of the merged graph has utility at least $\mathrm{PMMS}(V)$. We defer the missing details to the full paper. □

Proof of Theorem 4.1. Suppose that there are two agents with utility functions u_1, u_2. By Theorem 4.2, we can compute a (3/4)-PMMS connected allocation (X, Y) with respect to u_1 in polynomial time. Then, the allocation that assigns to agent 2 a preferred bundle among X and Y and the rest to agent 1 is (3/4)-PMMS. □

5 THE CASE OF THREE AGENTS

In this section, we prove that a PMMS allocation always exists and can be found in polynomial time for three agents with additive utilities on a path. While limited, this case is interesting as existence of PMMS remains open for three agents with additive utilities in the standard setting. As the proof relies on case analysis, it is deferred to the full paper. This section presents the method used in the proof.

Theorem 5.1. *For a path and three agents with additive utilities, a PMMS connected allocation always exists and can be found in polynomial time.*

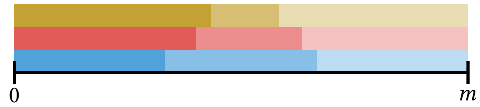
To find a PMMS connected allocation, we first find a PMMS partition for each of the three agents. This can be done in polynomial time by Theorem 6.5. Each of the three PMMS partitions can be represented by a pair of edges, namely the edge separating the left and middle bundle and the edge separating the middle and right bundle. Depending on the order in which these six edges (two for each agent) appear on the path, we can proceed in one of two ways.

In most cases, such as in Fig. 2a, a PMMS allocation can be found by simply allocating the bundles in one of the PMMS partitions to the agents in a certain way. For example, in Fig. 2a a PMMS allocation can be obtained by giving the left and right bundles in the topmost PMMS partition (yellow) to the two bottom agents (blue and red) in any way and the middle bundle to the top agent (yellow). This is guaranteed to be a PMMS allocation by the following lemma.

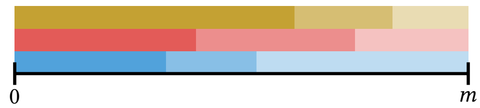
Lemma 5.2. *For a path and an additive utility function u, let (B_1, B_2) be a PMMS partition for u of the subpath given by $B_1 \cup B_2$. Let B_1' and B_2' be two neighbouring bundles in some partition of the path, such that $B_1' \subseteq B_1$ and $B_2 \subseteq B_2'$. Then, $u(B_2') \geq \mathrm{PMMS}(B_1' \cup B_2')$.*

Proof. Assume that $u(B_2') < \mathrm{PMMS}(B_1' \cup B_2')$. Since the graph is a path, there must then by the definition of PMMS exist $B' \subseteq B_1'$ such that $B' \cup B_2'$ and $B_1' \setminus B'$ are connected and $u(B_2') < \min\{u(B' \cup B_2'), u(B_1' \setminus B')\}$. In other words, the items in B' can be transferred from B_1' to B_2' to improve the utility of the worst bundle.

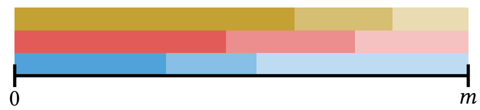
Let $B = B' \cup (B_2' \cap B_1)$. Since $B_1' \subseteq B_1$ it holds that $B \subseteq B_1$. Moreover, it must hold that $B_2 \cup B$ is connected as otherwise $B_2' \cup B'$ is not connected. Since $u(B' \cup B_2') > u(B_2')$ it holds that $u(B) \geq u(B') > 0$ and $u(B \cup B_2) > u(B_2)$. Moreover, $u(B_2) \leq u(B_2') < u(B_1' \setminus B') \leq u(B_1 \setminus B)$. Thus, $\min\{u(B_1), u(B_2)\} < \min\{u(B_1 \setminus B), u(B_2 \cup B)\}$ and (B_1, B_2) is not a PMMS partition of the subpath for u, a contradiction. Consequently, it holds that $u(B_2') \geq \mathrm{PMMS}(B_1' \cup B_2')$. □



(a) Simple case



(b) Difficult case



(c) Modified difficult case

Figure 2: Three possible layouts of PMMS partitions for 3 agents on a path.

In some cases, such as in Fig. 2b, it may be that in every one of the three PMMS partitions, there is only a single bundle that satisfies PMMS for each of the two other agents. If this is the same bundle for both agents, another method must be used to construct a PMMS allocation. In these few cases, it can be shown that one of the PMMS partitions is such that the middle bundle has the highest utility for both of the other agents. For example, in the case in Fig. 2b, this would be the case for the middle bundle of the middle agent. If this was not the case for the bottom (top) agent, then since the left (right) bundle covers more than an entire bundle in the agent's own PMMS partition, we can show that the left (right) bundle in the middle PMMS partition satisfies PMMS for the agent.

If both other agents think the middle bundle has the most utility, we can show that we can select one of the two and let her move the border between two of the bundles in the partition in a way that shrinks the middle bundle. Since the middle bundle had most utility, we can guarantee that the new partition will satisfy PMMS for the agent if she receives either of the modified bundles.

Lemma 5.3. *For a path and an additive utility function u, let (B_1, B_2, B_3) be a partition of the path into connected bundles where B_2 neighbours both B_1 and B_3, and $u(B_2) > u(B_1) \geq u(B_3)$. Then, there exists a PMMS partition (B_1', B_2') of $B_1 \cup B_2$ with $B_1 \subseteq B_1'$, $B_2' \subseteq B_2$ and $u(B_2') > u(B_3)$.*

Proof. If (B_1, B_2) is already a PMMS partition of $B_1 \cup B_2$, then all the properties are satisfied. Otherwise, we must have that $u(B_1) = \min\{u(B_1), u(B_2)\} < \mathrm{PMMS}(B_1 \cup B_2)$. Since the graph is a path, the only way of increasing the utility of the bundle B_1 is by moving items from B_2 to B_1. Therefore, any PMMS partition (B_1', B_2') of $B_1 \cup B_2$ must, up to naming, be such that $B_1 \subset B_1'$ and $B_2' \subset B_2$. Moreover, it follows that $u(B_2') \geq \mathrm{PMMS}(B_1 \cup B_2) > u(B_1) \geq u(B_3)$. Thus, any PMMS partition of $B_1 \cup B_2$ satisfies the properties. □

Notice that $u(B_2') > u(B_3)$. Thus, it must hold that $u(B_2') \geq u(B_2' \cup B_3)/2 \geq \mathrm{PMMS}(B_2' \cup B_3)$. Since (B_1', B_2') is a PMMS partition of $B_1' \cup B_2'$, both B_1' and B_2' must satisfy PMMS for u in the partition (B_1', B_2', B_3). Further notice that $B_1 \subseteq B_1'$, and $B_2' \subseteq B_2$. Since (B_1, B_2, B_3) was a PMMS partition for one of the agents, Lemma 5.2

can be used to show that B_1' and B_3 must satisfy PMMS for this agent in the new partition (B_1', B_2', B_3). Using a similar observation to that of Feige et al. [16], this guarantees that the bundles in this new partition can be given to the three agents in such a way that the resulting allocation is PMMS. Figure 2c shows an example of one such possible modification to the PMMS partition in Fig. 2b.

6 IDENTICAL UTILITIES

When agents have identical utilities, a PMMS and MMS connected allocation is guaranteed to exist for any graph. The proof is similar to that of Theorem 4.2 in Plaut and Roughgarden [30].

Theorem 6.1. *For a connected graph G and n agents with identical utilities, a PMMS and MMS connected allocation exists.*

Proof. Let A be a connected leximin allocation. Assume that A is not PMMS. Let i and j be a pair of agents for which PMMS is not satisfied and (A_i', A_j') a PMMS partition of $A_i \cup A_j$. Let A' be the allocation obtained by exchanging A_i and A_j for A_i' and A_j' in A. Then $u(A_k') = u(A_k)$ for every agent $k \in N \setminus \{i, j\}$, and $\min\{u(A_i'), u(A_j')\} > \min\{u(A_i), u(A_j)\}$. Thus, A is not leximin, a contradiction, and it follows that A must be PMMS. Moreover, A is MMS, as otherwise any MMS allocation, of which at least one exists for identical utilities, would be a leximin improvement of A. □

Theorem 6.1 shows that a leximin allocation is both MMS and PMMS for identical utilities, no matter the utility function and graph. For many combinations of graph classes and utility functions, finding a leximin allocation is hard. However, with monotone utilities on a path, a leximin allocation can be found in polynomial time [8]. This yields the following corollary.

Corollary 6.2. *For a path and n agents with identical monotone utilities, a PMMS and MMS connected allocation can be found in polynomial time.*

6.1 3/4-PMMS on General Graphs

Finding a leximin allocation is strongly NP-hard for identical utilities [18]. Unless P=NP, this excludes the existence of a pseudo-polynomial time algorithm for finding a PMMS connected allocation via a leximin allocation even when agents have identical utilities. We show on the other hand that there exists a pseudo-polynomial time algorithm for a (3/4)-PMMS connected allocation based on a sequence of local improvements between neighboring agents.

Theorem 6.3. *For a connected graph G and n agents with identical additive utility function* $u\colon V \longrightarrow \mathbb{Z}_+$, *a* $(3/4)$*-PMMS connected allocation can be found in pseudo-polynomial time.*

Proof sketch. Consider an algorithm that repeatedly finds a pair of agents whose bundles violate the PMMS condition and reallocating their bundles using Theorem 4.2 (formalized as Algorithm 2 in the full paper). One can show that over the course of the algorithm, the sum of squares $\sum_{i \in N}(u_i(A_i))^2$ decreases by at least 1 after a polynomial number of steps. Thus, the number of iterations is at most $nu_i(V)^2 \le n(nu_{max})^2$ where $u_{max} = \max_{v \in V} u(v)$. Further, by Theorem 4.2, each step of the **while**-loop can be implemented in polynomial time. This establishes the claim. □

Algorithm 2 Algorithm for constructing a PMMS and SMMS connected allocation for a connected graph $G = (V, E)$ and n agents with identical utilities u

1: Find a SMMS connected allocation A and create graph G' from G by removing the items allocated to losers in A.
2: **for** each connected component $C = (V_C, E_C)$ in G' **do**
3: Run Algorithm 2 for C and agents $N_C = \{ i \mid A_i \subseteq C, i \in [n] \}$, and let A_C be the resulting allocation.
4: For each $i \in N_C$, swap i's bundle in A for i's bundle in A_C.
5: **end for**
6: **return** A

6.2 PMMS on Trees

Forgoing leximin, we can construct an algorithm that guarantees both PMMS and MMS for identical utilities. For this purpose, we make use of the following strengthening of MMS introduced by Bilò et al. [8]. If agents have identical utilities, an allocation A is *strongly maximin share* (SMMS) if it is MMS and minimizes the number of agents i with $u(A_i) = \text{MMS}(V)$ among all MMS allocations. An agent i with $u(A_i) = \text{MMS}(V)$ is called a *loser*. Algorithm 2 finds a PMMS and MMS connected allocation by finding an SMMS allocation, fixing the bundles of the losers and repeating the process for the remaining agents. Since SMMS is a stronger requirement than MMS, Algorithm 2 is not polynomial in the general case, unless P=NP. However, for trees and additive utilities, we can show that the algorithm is polynomial. Note that the output of the algorithm may not be leximin since PMMS and SMMS may not imply leximin; See Lemma 6.6 in the full paper.

Lemma 6.4. *For a connected graph G and n with identical utility function u, Algorithm 2 finds a PMMS and SMMS connected allocation. It runs in polynomial time if connected SMMS allocations can be found in polynomial time and the utility* $u(X)$ *can be computed in polynomial time for each* $X \subseteq V$.

Proof. We prove PMMS and SMMS by induction on recursive calls to Algorithm 2. When every agent is a loser in A—our inductive base case—A must be both SMMS and leximin, as any improvement would result in fewer losers. By Theorem 6.1, A is thus PMMS.

For our inductive step, we claim that if each A_C found is a PMMS and SMMS connected allocation, then A is a PMMS and SMMS connected allocation when returned on Line 6. Since an SMMS connected allocation is found on Line 1, every non-loser agent belongs to exactly one set N_C and A remains a connected allocation. Also, $\mu^{|N_C|}(V_C) > \mu^n(V)$ must hold, as $u(A_i) > \mu^n(V) = \text{MMS}(V)$ for every $i \in N_C$. Thus, every agent i that was not a loser on Line 1 receives a bundle with $u(A_i) > \text{MMS}(V)$ and A remains SMMS.

Since A_C is PMMS, the only way PMMS can not hold in A is between a loser i and an agent $j \in N_C$ for some connected component C. Assume this is the case when A is returned and let (B_i, B_j) be a PMMS partition of $A_i \cup A_j$. Replacing A_i and A_j by B_i and B_j would reduce the number of losers in A or increase the utility of the worst off agent, as $\min\{u(A_i), u(A_j)\} < \min\{u(B_i), u(B_j)\}$. This is a contradiction, as then A is not SMMS. Hence, A is PMMS.

Since any SMMS allocation has at least one loser and every non-loser belongs to exactly one set N_C, Algorithm 2 is ran at most n times. Under the stated assumptions, we can easily verify that every operation in the algorithm can be performed in polynomial time. Thus, the algorithm must run in polynomial time. □

We now show that an SMMS connected allocation can be found for identical additive utilities on trees in polynomial time.

Theorem 6.5. *For a tree G and n agents with identical additive utilities, a SMMS connected allocation can be found in polynomial time.*

Finding an SMMS allocation is trivial if $\mathrm{MMS}(V) = 0$; Simply give as many agents as possible items with non-zero utility. When $\mathrm{MMS}(V) > 0$, we solve a related problem: First, root the tree in some arbitrary vertex r. For any vertex v_i and pair of integers $0 \le j, \ell \le n$, we are interested in finding, if it exists, a partition $P_{i,j,\ell} = (B_1, \ldots, B_{j+1})$ of the subtree rooted in v_i into connected bundles that subject to the following conditions maximizes $u(B_{j+1})$:

(i) $B_{j+1} = \emptyset$ or $v_i \in B_{j+1}$
(ii) $u(B_t) \ge \mathrm{MMS}(V)$ for $1 \le t \le j$
(iii) $|\{ B_t \mid u(B_t) = \mathrm{MMS}(V), 1 \le t \le j \}| \le \ell$

In other words, $P_{i,j,\ell}$ partitions the subtree into j bundles with utility at least $\mathrm{MMS}(V)$ of which at most ℓ have a utility of exactly $\mathrm{MMS}(V)$. Finally, the remaining bundle, B_{j+1}, is either empty or contains v_i. This property is important, as B_{j+1}, the only bundle not guaranteed to have a utility of at least $\mathrm{MMS}(V)$, can always be combined with a bundle containing v_i's parent. Together with maximizing $u(B_{j+1})$, this allows for solutions to be found by dynamic programming. Note that since G is a tree, $\mathrm{MMS}(V)$ can be found in polynomial time using an algorithm of Perl and Schach [29]. We now show that solving the problem for $i = r$, $j = n$ and $0 \le \ell \le n$ yields an SMMS connected allocation.

Lemma 6.6. *For a tree $G = (V, E)$ rooted in some vertex $r \in V$ and n agents with identical utilties, $P_{r,n,\ell}$ exists for some $0 \le \ell \le n$ and a SMMS connected allocation of G can be obtained from $P_{r,n,\ell^*} = (B_1, \ldots, B_n, B_{n+1})$ by removing B_{n+1} and redistributing the items in B_{n+1} to the other bundles in any valid way, where ℓ^* is the smallest $0 \le \ell \le n$ for which $P_{r,n,\ell}$ exists.*

Proof. Let $A = (A_1, \ldots, A_n)$ be an SMMS connected allocation for G and $\ell_A = |\{ A_i \mid u(A_i) = \mathrm{MMS}(V), i \in [n] \}|$. We claim that $A' = (A_1, \ldots, A_n, \emptyset)$ satisfies (i)–(iii) for the triple r, n, ℓ_A. Indeed $B_{j+1} = \emptyset$ and (i) holds. Since A is SMMS, (ii) holds. By the definition of ℓ_A, (iii) also holds. Thus, $P_{r,n,\ell}$ exists for at least one ℓ.

Let $B = (B'_1, \ldots, B'_n)$ be the allocation obtained from P_{r,n,ℓ^*} by removing B_{n+1} and redistributing the items. By (ii), every bundle $B'_i \in B$ has $u(B'_i) \ge \mathrm{MMS}(V)$. Since ℓ^* is minimized, it holds that $\ell^* \le \ell_A$ and by (iii) the number of bundles with utility exactly $\mathrm{MMS}(V)$ is no greater in B than in A. In fact, it must hold that $\ell^* = \ell_A$ as otherwise A is not SMMS. Thus, B must be SMMS. □

To solve the problem for the selected root r, $j = n$ and $0 \le \ell \le n$, we will rely on dynamic programming to combine solutions for the children of r.

Lemma 6.7. *For a vertex v_i, $P_{i,j,\ell}$ can be computed in polynomial time for any $0 \le j, \ell \le n$ if P_{h,j_h,ℓ_h} is known for every child vertex v_h of v_i and pair $0 \le j_h, \ell_h \le n$.*

Proof sketch of Lemma 6.7. If v_i is a leaf, then the subtree consists of a single vertex and there are two possible partitions that can satisfy (ii): $(\{v_i\})$ and $(\{v_i\}, \emptyset)$. For any pair j and ℓ, (i)–(iii) can be checked for these two partitions in polynomial time.

We now consider internal vertices of the tree. Assume that some solution exists for P_{h,j_h,ℓ_h}, and fix this solution. Then, for every child v_h of v_i there is some number s_h of bundles in which items from the subtree rooted in v_h appear. Since at most one of these bundles, the one containing v_h can contain other items from the graph, it can be shown that these bundles in $P_{i,j,\ell}$ can be replaced by a solution for $P_{h,(s_h-1),\ell_h}$ where ℓ_h is the number of these bundles in $P_{i,j,\ell}$ that have utility exactly $\mathrm{MMS}(V)$. Consequently, if a solution exists, there is some solution that consists of solutions to the problem for each of the child vertices.

To construct a solution or determine that a solution does not exist, one can try every combination of solutions for the children. Either no combination gives a solution or the one that maximizes the utility of B_{j+1} can be selected. Directly testing all combinations can require exponential time if v_i has a large number of children. However, as utilities are additive and a maximum is to be found, a solution can be found iteratively by first combining solutions for two children and selecting ones that provide a maximal contribution in utility to B_{j+1}. Then, the process can be repeated, combining the partial solutions with solutions for the next child until there is no child of v_i that has not been considered. □

Proof of Theorem 6.5. By Lemma 6.7 we can compute $P_{i,j,\ell}$ in polynomial time for any vertex v_i and pair j, ℓ if we have solved the problem for every child of v_i. Thus, $P_{i,j,\ell}$ can be computed through a post-order traversal of the tree. Since there is a polynomial number, $|V|(n+1)^2$, of solutions to compute, we can determine all in polynomial time. Given $P_{r,n,\ell}$ for $0 \le \ell \le n$, an SMMS connected allocation can be constructed in polynomial time using Lemma 6.6. □

Corollary 6.8. *For a tree G and n agents with identical additive utilities, a PMMS and SMMS connected allocation can be found in polynomial time.*

With an approach similar to that of Truszczynski and Lonc [33] for MMS in unicyclic graphs—graphs that contain at most a single cycle—Theorem 6.5 and Corollary 6.8 can be extended to unicyclic graphs (see the full paper). Unfortunately, it is unlikely that polynomial time algorithms for SMMS can be found in most cases with more complex graph classes or utility functions.[5]

7 CONCLUSION

We introduced a novel local adaptation of PMMS into graph fair division. Our work made a major first step in addressing critical questions related to the existence and algorithmic aspects of PMMS within this context. We leave several important questions open. Most notably, the existence of a PMMS connected allocation is inevitably a challenging open problem due to the fact that this problem is at least as difficult as in the standard setting of fair division. Another direction is to consider a “local" variant of other solution concepts, such as MMS. For instance, instead of pairwise comparison, agents may compare their bundles with all the bundles of their neighbors.

[5] See Lemmas 6.17 and 6.18 in the full paper.

REFERENCES

[1] Rediet Abebe, Jon Kleinberg, and David C. Parkes. 2017. Fair Division via Social Comparison. In *Proceedings of the 16th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS).* 281–289.

[2] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A. Voudouris, and Xiaowei Wu. 2022. Fair Division of Indivisible Goods: Recent Progress and Open Questions. arXiv:2208.08782 [cs.GT]

[3] Georgios Amanatidis, Georgios Birmpas, and Vangelis Markakis. 2018. Comparing Approximate Relaxations of Envy-Freeness. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI).* 42–48.

[4] Georgios Amanatidis, Evangelos Markakis, and Apostolos Ntokos. 2020. Multiple birds with one stone: Beating 1/2 for EFX and GMMS via envy cycle elimination. *Theoretical Computer Science* 841 (2020), 94–109.

[5] Xiaohui Bei, Ayumi Igarashi, Xinhang Lu, and Warut Suksompong. 2022. The Price of Connectivity in Fair Division. *SIAM journal on Discrete Mathematics* 36, 2 (2022), 1156–1186.

[6] Xiaohui Bei, Youming Qiao, and Shengyu Zhang. 2017. Networked Fairness in Cake Cutting. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI).* 3632–3638.

[7] Aurélie Beynier, Yann Chevaleyre, Laurent Gourvès, Julien Lesca, Nicolas Maudet, and Anaëlle Wilczynski. 2019. Local envy-freeness in house allocation problems. *Autonomous Agents and Multi-Agent Systems* 33 (2019), 591–627.

[8] Vittorio Bilò, Ioannis Caragiannis, Michele Flammini, Ayumi Igarashi, Gianpiero Monaco, Dominik Peters, Cosimo Vinci, and William S. Zwicker. 2022. Almost Envy-Free Allocations with Connected Bundles. *Games and Economic Behavior* 131 (2022), 197–221.

[9] John Adrian Bondy and U. S. R. Murty. 2008. *Graph Theory* (1st ed.). Springer.

[10] Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, and Dominik Peters. 2017. Fair Division of a Graph. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI).* 135–141.

[11] Steven J. Brams and Peter C. Fishburn. 2000. Fair Division of Indivisible Items Between Two People with Identical Preferences: Envy-Freeness, Pareto-Optimality, and Equity. *Social Choice and Welfare* 17, 2 (2000), 247–267.

[12] Steven J. Brams, Marc Kilgour, and Christian Klamler. 2014. Two-Person Fair Division of Indivisible Items: An Efficient, Envy-Free Algorithm. *Notices of the AMS* 61, 2 (2014), 130–141.

[13] Robert Bredereck, Andrzej Kaczmarczyk, and Rolf Niedermeier. 2022. Envy-free allocations respecting social networks. *Artificial Intelligence* 305 (2022), 103664.

[14] Eric Budish. 2011. The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes. *Journal of Political Economy* 119, 6 (2011), 1061–1103.

[15] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. 2019. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Transactions on Economics and Computation* 7, 3 (2019), 12:1–12:32.

[16] Uriel Feige, Ariel Sapir, and Laliv Tauber. 2022. A Tight Negative Example for MMS Fair Allocations. In *Web and Internet Economics (Lecture Notes in Computer Science)*, Michal Feldman, Hu Fu, and Inbal Talgam-Cohen (Eds.). Springer International Publishing, Cham, 355–372. https://doi.org/10.1007/978-3-030-94676-0_20

[17] Leon Festinger. 1954. A theory of social comparison processes. *Human Relations* 7 (1954), 117–140.

[18] Michael R. Garey and David S. Johnson. 1979. *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., USA.

[19] Gianluigi Greco and Francesco Scarcello. 2020. The Complexity of Computing Maximin Share Allocations on Graphs. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI).* 2006–2013.

[20] Hadi Hosseini, Justin Payan, Rik Sengupta, Rohit Vaish, and Vignesh Viswanathan. 2023. Graphical House Allocation. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS).* 161–169.

[21] Ayumi Igarashi. 2023. How to cut a discrete cake fairly. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI).*

[22] Ayumi Igarashi and Dominik Peters. 2019. Pareto-Optimal Allocation of Indivisible Goods with Connectivity Constraints. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 33. 2045–2052.

[23] D. Marc Kilgour and Rudolf Vetschera. 2018. Two-Player Fair Division of Indivisible Items: Comparison of Algorithms. *European Journal of Operational Research* 271, 2 (2018), 620–631.

[24] David Kurokawa. 2017. *Fair division in game theoretic settings.* Ph.D. Dissertation. Carnegie Mellon University.

[25] David Kurokawa, Ariel D. Procaccia, and Junxing Wang. 2018. Fair enough: Guaranteeing approximate maximin shares. *J. ACM* 64, 2 (2018), 8:1–8:27.

[26] Abraham Lempel, Shimon Even, and Israel Cederbaum. 1967. An algorithm for planarity testing of graphs. *Theory of Graphs: International Symposium* (1967), 215–232.

[27] Zbigniew Lonc. 2023. Approximating Fair Division on D-Claw-Free Graphs. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence.* International Joint Conferences on Artificial Intelligence Organization, Macau, SAR China, 2826–2834. https://doi.org/10.24963/ijcai.2023/315

[28] Hervé Moulin. 2003. *Fair Division and Collective Welfare.* MIT Press.

[29] Yehoshua Perl and Stephen R. Schach. 1981. Max-Min Tree Partitioning. *J. ACM* 28, 1 (Jan. 1981), 5–15. https://doi.org/10.1145/322234.322236

[30] Benjamin Plaut and Tim Roughgarden. 2020. Almost Envy-Freeness with General Valuations. *SIAM Journal on Discrete Mathematics* 34, 2 (Jan. 2020), 1039–1068.

[31] Leonard I. Schieman, Scott Pearlin. 2006. Neighborhood Disadvantage, Social Comparisons, and the Subjective Assessment of Ambient Problems Among Older Adults. *Social Psychology Quarterly* 69, 3 (2006), 253–269.

[32] Jerry Suls and Ladd Wheeler (Eds.). 2000. *Handbook of Social Comparison: Theory and Research.* Springer New York, NY.

[33] Miroslaw Truszczynski and Zbigniew Lonc. 2020. Maximin Share Allocations on Cycles. *Journal of Artificial Intelligence Research* 69 (Oct. 2020), 613–655.