Pareto-Optimal Allocation of Indivisible Goods with Connectivity Constraints

Ayumi Igarashi[1] and Dominik Peters[2]

[1] Kyushu University, Fukuoka, Japan igarashi@agent.inf.kyushu-u.ac.jp
[2] University of Oxford, Oxford, U.K. dominik.peters@cs.ox.ac.uk

Abstract. We study the problem of allocating indivisible items to players with additive utilities, under the additional constraint that bundles must be connected in an underlying item graph. Previous work has considered the existence and complexity of fair allocations. We study the problem of finding an allocation that is Pareto-optimal. While it is easy to find an efficient allocation when the underlying graph is a path or a star, the problem is NP-hard for many other graph topologies, even for trees of bounded pathwidth. We also show that it is NP-hard to find a Pareto-optimal allocation that satisfies maximin share even on a path.

1 Introduction

We consider divisions of a set of indivisible items among multiple players. Specifically, given a set of items and a set of agents who have different preferences over the items, we are interested in finding an allocation that satisfies certain fairness desiderata. Although defining fairness itself is a tricky problem, there are several established notions of fairness including envy-freeness and proportionality. Recently introduced are *maximin fair share* (MMS) [11] and *envy-freeness up to one good* (EF1) [16], which have attracted a considerable amount of attention in the recent literature [9, 12, 6].

The standard approach to fair division of indivisible allocations imposes no constraints on possible combinations of items. However, real-world applications often involve restrictions on feasible bundles; for instance, in fair allocation of land plots, one needs to divide the plots into disjoint connected sets among agents. Recently, Bouveret et al. [9] considered fair allocation of indivisible items under a graph-connectivity constraint: there is an undirected graph describing the relationship between the items, and each agent's share must form a connected subgraph of this graph. Examples include the preceding example of land division, time-scheduling, to name a few.

One of most prominent results in this direction is perhaps the result of Bouveret et al. [9], showing that an allocation satisfying maximin fair share always exists and is polynomial time-computable if the network is acyclic. Their algorithm corresponds to a discrete version of the Dubins–Spanier moving-knife procedure ensuring proportionality while cutting a continuous cake.

Nevertheless, fairness is not the only criterion in achieving a desirable division of goods. For example, consider an instance of two agents with two items where each agent approves a different item. While giving a non-approved item to each agent is arguably fair, such an allocation is not efficient. Indeed, much of the recent literature has focused on whether multiple normative properties, such as fairness and efficiency, can be satisfied under certain conditions on the inputs; in particular, identifying where Pareto-optimality and fairness can be simultaneously achieved has been the subject of intense research in the (computational) social choice literature [11, 12, 6].

Our contribution

In this paper, we consider the problem of dividing indivisible objects from the efficiency perspective. The most commonly used notion of efficiency is *Pareto-optimality*. An allocation is *Pareto-optimal* if there is no reallocation of goods in a way where at least one individual would be better off while no other individual ends up worse off. In contrast with the non-constrained setting where finding a Pareto-optimal allocation is polynomial-time solvable, we show that the respective problem is NP-hard, even for trees of bounded pathwidth and with binary preferences. We also investigate the computational complexity of finding an efficient allocation while ensuring fairness. While a Pareto-optimal and MMS allocation always exist on trees, we show that finding one is hard even for paths. For Pareto-optimality and EF1, we observe that even when players' approval items form a connected segment on paths, an allocation satisfying both properties may not exist; further we show that deciding the existence of such an allocation is NP-hard. We summarize our complexity results in Table 1.

	general	cliques	trees	paths
MMS			poly time [9]	polytime
PO	NP-h*	poly time	NP-h.* (Thm. 4)	poly time (Thm. 1)
PO & MMS	NP-h*		NP-h.*	NP-h.* (Thm. 6)
PO & EF1	NP-h.	poly time	NP-h.	NP-h. (Thm. 5)

Table 1. Overview of our complexity results. Hardness results marked * hold under Turing reductions. When no reference is given, the result follows directly from other results in the table. All hardness results hold even when utilities are binary.

Related Work

There is a rich body of the literature on fair division of *divisible* items with connected pieces. It is well known that an envy-free allocation of divisible items with contiguous pieces always exists [17]; nevertheless, it cannot be obtained in finite steps even when the cake is divided among three players [18]. In contrast, several efficient algorithms are known to yield a contiguous proportional allocation; see the survey by Lindner and Rothe [15] for more details.

The relation between efficiency and fairness under contiguity requirement is also well-understood in the context of allocation of divisible items. Aumann and Dombb [2] studied the efficiency loss in order to achieve fair allocations under connectivity constraints. Closely related to ours are the papers by Bei et al. [7] and Aumann et al. [3], who considered the computational complexity of finding a utilitarian optimal allocation with connected pieces. Bei et al. [7] showed the inapproximability of maximizing efficiency conditioned on proportionality; however, without the proportionality requirement, Aumann et al. [3] showed that there is an efficient constant approximation algorithm for finding an utilitarian optimal allocation. The algorithm in [3] is discrete and applies to our setting when the graph is a path.

In the context of division of indivisible items, Bouveret et al. [9] were the first to formalize the model of fair division with the extra feature that each bundle needs to be connected in the underlying item graph. While they showed that finding an envy-free or proportional connected allocation is NP-hard even on paths, they proved that an allocation satisfying the maximin fair share always exists and can be found in polynomial time when the graph does not admit cycles. Independently of their work, Suksompong [19] considered the problem when the items lie on a line, obtaining a number of approximation guarantees with respect to several fairness notions such as envy-freeness and proportionality.

Finally, Aziz et al. [4] obtained a number of complexity results for the problem of finding Pareto-improvements of a given allocation when players have additive preferences. We note that their result is for the case where the item graph is a clique, whereas most of our results are for the case when the graph is a tree.

2 Preliminaries

For each natural number $s \in \mathbb{N}$, we write $[s] = \{1, 2, \ldots, s\}$. Given a graph $G = (V, E)$ and subset X of the vertices, we denote by $G \setminus X$ the subgraph of G induced by $V \setminus X$.

We will define the connected fair division (CFD) problem, introduced by Bouveret et al. [9]. An *instance* of CFD is given by a triple $I = (G, N, \mathcal{U})$ where $G = (V, E)$ is an undirected graph, $N = [n]$ is a finite set of *players*, and a *utility profile* $\mathcal{U}$ is an n-tuple of non-negative utility functions $u_i : V \to \mathbb{R}_+$. We refer to elements of V as *items*, and denote the number of items by m. A utility profile $\mathcal{U}$ is said to be *binary* if for each $i \in N$ and each $v \in V$, $u_i(v) \in \{0, 1\}$; we say that player i *approves* an item v if $u_i(v) = 1$. Our positive algorithmic results hold even if utilities are encoded in binary, and our hardness results hold even if utilities are encoded in unary (since all of them hold even for binary utilities).

Subsets $X \subseteq V$ are referred to as *bundles* of items. For each bundle $X \subseteq V$, we write $u_i(X) = \sum_{v \in X} u_i(v)$. An *allocation* is a function $\pi : N \to 2^V$ assigning each player a bundle of items, such that each item is allocated at most once, i.e., $\pi(i) \cap \pi(j) = \emptyset$ for each pair of distinct players $i, j \in N$. An allocation π is *connected* if for each player $i \in N$ the bundle $\pi(i)$ is connected in G.

Given an allocation π, another allocation π' is a *Pareto-improvement* of π if $u_i(\pi'(i)) \geqslant u_i(\pi(i))$ for all $i \in N$ and $u_j(\pi'(j)) > u_j(\pi(j))$ for some $j \in N$. We say that a connected allocation π is *Pareto-optimal* (or *efficient*, or *PO* for short) if there is no connected allocation that is a Pareto-improvement of π. The *utilitarian social welfare* of an allocation π is $\sum_{i \in N} u_i(\pi)$. It is easy to see that a connected allocation which maximizes utilitarian social welfare among connected allocations is Pareto-optimal.

The standard fairness notions of *maximin share (MMS) allocations* [11] and *envy-freeness up to one good* (EF1) [12] can be adapted to the graph-restricted setting as follows. Given an instance $I = (G, N, \mathcal{U})$ of CFD with $G = (V, E)$, let Π_n denote the space of all partitions of V into n connected pieces. The *maximin share guarantee* of a player $i \in N$ is

$$\mathrm{mms}_i(I) = \max_{(P_1,\ldots,P_n) \in \Pi_n} \min_{j \in [n]} u_i(P_j).$$

A connected allocation π is a *maximin share (MMS) allocation* if $u_i(\pi(i)) \geqslant \mathrm{mms}_i(I)$ for each player $i \in N$. We say that a connected allocation satisfies EF1 if the envy is bounded up to an *outer* good, i.e., for any pair of players $i, j \in N$, either $u_i(\pi(i)) \geqslant u_i(\pi(j))$ or there is a good $v \in \pi(j)$ such that $u_i(\pi(i)) \geqslant u_i(\pi(j) \setminus \{v\})$ and $\pi(j) \setminus \{v\}$ remains connected.

Finally we introduce several measures of a graph that are often used to represent the structural complexity of the associated problem. The *diameter* of a graph G is the maximum distance between any pair of vertices. A *path decomposition* of a graph G is a sequence $(X_1, X_2, \ldots, X_k)$ of subsets of vertices of G such that the endpoints of each edge appear in one of the subsets and that each vertex appears in a contiguous subsequence of the subsets; the *pathwidth* of a path decomposition is $\max_{i \in [k]} |X_i| - 1$ and the *pathwidth* of a graph G is the minimum pathwidth taken over all possible path decompositions of G.

3 Finding Any Pareto-Optimal Allocation

We start by considering the problem of producing *some* Pareto-optimal allocation, without imposing any additional constraints on the quality of this allocation. In the standard setting where there are no connectivity requirements (equivalently, when G is the complete graph), this problem is trivial: Simply allocate each item separately to an agent who has a highest valuation for that item. The resulting allocation maximizes utilitarian social welfare and is thus Pareto-optimal. When G is not complete, the problem with this approach is that it can produce disconnected bundles. Is it still possible to find an efficient allocation for other graph topologies?

3.1 Paths and Stars

For very simple graphs, the answer is positive. When G is a path, a certain serial dictatorship method produces a Pareto-optimum.

Theorem 1. *When G is a path, a Pareto-optimal allocation can be found in polynomial time.*

Proof. The path G is given by $V = \{v_1, v_2, \dots, v_m\}$ where $\{v_j, v_{j+1}\} \in E$ for each $j \in [m-1]$. For a subset V' of V, we denote by $\min V'$ the vertex of minimum index among the vertices in V'. For an allocation π and a subset N' of players, we denote by $\pi|_{N'}$ the allocation restricted to N'.

We design a recursive algorithm $\mathcal{A}$ that takes as input a subset N' of players, a subpath $G' = (V', E')$ of G, and a utility profile $(u_i)_{i \in N'}$, and returns a Pareto-optimal allocation of the items in V' to the players in N'. Without loss of generality, we may assume that there is a player who likes the left-most vertex of G', i.e., $u_i(\min V') > 0$ for some $i \in N'$, since otherwise we can remove that vertex from the graph without changing the set of Pareto-optimal allocations.

If $|N'| = 1$, then the algorithm allocates all items V' to the single player. Suppose that $|N'| > 1$. The algorithm first finds a player i who has positive utility for $\min V'$; it then allocates to i a minimal connected bundle $V_i \subseteq V'$ containing all items in V' to which i assigns positive utility (so that $u_i(V_i) = u_i(V')$). To decide on the allocation of the remaining items, we apply $\mathcal{A}$ to the reduced instance $(N' \setminus \{i\}, G' \setminus V_i, (u_{i'})_{i' \in N' \setminus \{i\}})$.

We will prove by induction on $|N'|$ that the allocation produced by $\mathcal{A}$ is Pareto-optimal. The is clearly true when $|N'| = 1$. Suppose that $\mathcal{A}$ returns a Pareto-optimal allocation when $|N'| = k-1$; we will prove it for $|N'| = k$. Let π be the allocation returned by $\mathcal{A}$, where $\mathcal{A}$ chose player i and allocated the bundle V_i. Assume for a contradiction that there is a Pareto-improvement π' of π. Thus, in particular, $u_i(\pi'(i)) \geqslant u_i(\pi(i))$. Hence, by the algorithm's choice of the bundle V_i, we must have $V_i \subseteq \pi'(i)$ and $u_i(\pi'(i)) = u_i(\pi(i))$. Hence, there must be a player j' different from i who receives strictly higher utility in π' than in π.

Now, since $G \setminus \pi'(i)$ is a subgraph of $G \setminus V_i$, the allocation $\pi'|_{N' \setminus \{i\}}$ is an allocation for the instance $I' = (N' \setminus \{i\}, G' \setminus V_i, (u_{i'})_{i' \in N' \setminus \{i\}})$. Also, we have

- $u_j(\pi'(j)) \geqslant u_j(\pi(j))$ for all players $j \in N' \setminus \{i\}$; and
- $u_{j'}(\pi'(j')) > u_{j'}(\pi(j'))$ for some $j' \in N' \setminus \{i\}$.

Thus, $\pi'|_{N' \setminus \{i\}}$ is a Pareto-improvement of the allocation $\pi|_{N' \setminus \{i\}}$. But $\pi|_{N' \setminus \{i\}}$ is the allocation returned by $\mathcal{A}$ for the instance I', contradicting the inductive hypothesis that $\mathcal{A}$ returns Pareto-optimal allocations for $|N'| = k-1$. □

Another graph type for which we can find a Pareto-optimum is a star; in fact, we can efficiently calculate an allocation maximizing utilitarian social welfare. Note that when G is a star, at most one player can receive two or more items. This allows us to translate welfare maximization into a bipartite matching instance.

Theorem 2. *When G is a star, a Pareto-optimal allocation can be found in polynomial time.*

Proof. We will show that finding a connected allocation that maximizes the utilitarian welfare is polynomial-time solvable on stars. To this end, we guess a player $i \in N$ who receives the central vertex c of the star. To decide how the leaf items are allocated, we construct a weighted bipartite graph H_i with bipartition $(N', V \setminus \{c\})$ where N' consists of players $j \in N \setminus \{i\}$ together with $m-1$ copies $i_1, i_2, \ldots, i_{m-1}$ of player i. (These copies represent 'slots' in i's bundle.) Each edge of form $\{j, v\}$ for $j \in N \setminus \{i\}$ has weight $u_j(v)$ and each edge of form $\{i_k, v\}$ has weight $u_i(v)$.

Observe that each connected allocation in which i obtains c can be identified with a matching in H_i: Every leaf object is either matched with the agent receiving it, or is matched with some copy i_k of i if the object is part of i's bundle. Note that utilitarian social welfare of this allocation equals the total weight of the matching. Since one can find a maximum-weight matching in a bipartite graph in polynomial time [14], we can find an allocation of maximum utilitarian welfare efficiently. $\square$

3.2 Hardness Results

For more general classes of graphs, the news is less positive. We will show that, unless P = NP, there is no polynomial-time algorithm which produces a Pareto-optimal allocation when G is an arbitrary tree, even for instances where G has bounded pathwidth, and where players' utilities are binary. In particular, note that this result implies that it is hard to find allocations maximizing any type of social welfare (utilitarian, leximin, Nash) when G is a tree, since such allocations are also Pareto-optimal.

To obtain our hardness result, we first consider a more general problem which is easier to handle, namely the case where G is a forest. Aziz et al. [5] showed that NP-hardness of finding a perfect partition implies NP-hardness of finding a Pareto-optimal partition in the context of hedonic coalition formation games. In a similar spirit, we will first show that checking the existence of a perfect allocation is NP-hard even for forests where the size of each connected component is at most three; we then consider the problem of finding a Pareto-optimal allocation for arbitrary trees. A connected allocation π is said to be *perfect* if each player receives a most-preferred connected bundle, that is, $u_i(\pi(i)) = \max\{\, u_i(X) \mid X \subseteq V \text{ is connected in } G \,\}$ for all $i \in N$.

Theorem 3. *It is* NP*-complete to determine whether a perfect connected allocation exists when G is a forest with maximum connected component of size at most* 3*, even if utilities are binary.*

Proof. The problem is in NP, since we can easily find the utility value that each player would need to receive in a perfect allocation (by considering each connected component of G), and then a perfect allocation gives a certificate. Our hardness reduction is similar to a construction in [9] showing hardness of deciding the existence of a proportional allocation when G is a path. We reduce from Exact-3-Cover (X3C). Recall that an instance of X3C is given by a set of elements $X = \{x_1, x_2, \ldots, x_{3r}\}$ and a family $\mathcal{S}$ of three-element subsets of X; it is a 'yes'-instance if and only if there is an *exact cover* $\mathcal{S}' \subseteq \mathcal{S}$, i.e., $|\mathcal{S}'| = r$ and $\bigcup_{S \in \mathcal{S}'} S = X$.

Consider an instance $(X, \mathcal{S})$ of X3C. For each $S \in \mathcal{S}$, we denote the three elements of S by x_S^1, x_S^2, x_S^3. We construct an instance I of CFD as follows.

Items: For each set $S \in \mathcal{S}$, we create a path of three vertices x_S^1, x_S^2, x_S^3 with the center x_S^2; for each $k \in [r]$, we create a path of three dummy vertices d_k^1, d_k^2, d_k^3 with the center d_k^2. See Figure 1 for an illustration.

Fig. 1. Graph constructed in the proof of Theorem 3.

Players: For each set $S \in \mathcal{S}$, we create one player i_S who only approves the elements x_S^1, x_S^2, x_S^3 contained in the set, and each dummy d_k^h for $k \in [r]$ and $h \in [3]$; for each element $x \in X$ we create one player i_x who only approves the vertices x_S^h with $x_S^h = x$. Note that by construction, each player i_S $(S \in \mathcal{S})$ can receive a bundle of value at most 3, while each player i_x $(x \in X)$ can receive a bundle of value at most 1.

Correctness: We will show that I admits a perfect connected allocation if and only if there is an exact cover.

Suppose that there is a cover $\mathcal{S}'$ of size r. Then, one can construct a perfect connected allocation π as follows.

- Each player i_S whose corresponding set S appears in the cover $\mathcal{S}'$ is arbitrarily assigned to one of the the dummy triples d_k^1, d_k^2, d_k^3 for $k \in [r]$.
- Each player i_S whose corresponding set S does not appear in the cover $\mathcal{S}'$ is assigned to the triple x_S^1, x_S^2, x_S^3.
- Each player i_x is assigned to the element vertex x_S^k such that $x = x_S^k$ and $S \in \mathcal{S}'$.

Each player is assigned to a connected piece of maximum value, implying that π is a perfect connected allocation.

Conversely, suppose that there exists a connected allocation such that each player i_S receives a bundle of value 3 and each player i_x receives a bundle of value 1. Now, let $\mathcal{S}' \subseteq \mathcal{S}$ be the family of the sets $S \in \mathcal{S}$ where one of its elements x_S^1, x_S^2, x_S^3 is allocated to some x-players i_x. Since each player i_x needs to be allocated an item which she approves, $\mathcal{S}'$ forms a cover for X and hence $|\mathcal{S}'| \geqslant r$. On the other hand, as the number of dummy triples is r, there must be at least $|\mathcal{S}| - r$ players i_S who need to be allocated triple intervals x_S^1, x_S^2, x_S^3. Thus, $\mathcal{S}'$ has size at most r and constitutes an exact cover for X. $\square$

Corollary 1. *Unless* P = NP*, there is no polynomial time algorithm which finds a Pareto-optimal connected allocation when G is a forest, even if utilities are binary.*

Proof. Suppose there was such an algorithm. We show that we can then solve the problem of Theorem 3 in polynomial time: given an instance of that problem, use our algorithm to find a Pareto-optimal allocation, and return 'yes' if and only if that allocation is perfect. To see that this procedure correctly decides the existence of a perfect allocation, observe that *if* a perfect allocation exists, then all Pareto-optimal allocations are perfect. □

Building on the above reduction, we find that it is also hard to find an efficient allocation when G is a tree.

Theorem 4. *Unless* P = NP*, there is no polynomial time algorithm which finds a Pareto-optimal connected allocation when G is a tree, even if utilities are binary.*

Proof. To extend the reduction in the proof of Theorem 3 to trees, we first 'double' the reduction, in making a copy of each object and a copy of each player with the same preference as the original player. Specifically, given an instance $(X, \mathcal{S})$ of X3C, we create the same instance of CFD as in the proof of Theorem 3. In addition, we make a path of copies $\hat{x}_S^1, \hat{x}_S^2, \hat{x}_S^3$ of each triple x_S^1, x_S^2, x_S^3, together with a path of copies $\hat{d}_k^1, \hat{d}_k^2, \hat{d}_k^3$ of each triple d_k^1, d_k^2, d_k^3. We then make a copy $\hat{i}_S$ of each player i_S and a copy $\hat{i}_x$ of each player i_x. We also introduce a new dummy item c which serves as the center of a tree; specifically, we attach the center to one end of each of the connected components. The resulting graph G is a tree consisting of $2r + 2|\mathcal{S}|$ paths of length 3, each attached to the vertex c. No player has positive utility for the center dummy item. Copied players only value copied objects and have the same preferences as the corresponding original players, and non-copied players only value non-copied objects.

We will show that by computing a Pareto-optimal outcome of the modified instance, one can efficiently decide the existence of an exact cover. To see this, let π be a Pareto-optimal outcome. Then at most one player obtains the center item c and hence at most one player obtains a bundle that contains more than one 'leg', a connected component of the graph when deleting the center item c; without loss of generality, suppose it is a copied player.

Then the allocation π restricted to the non-copied players is a perfect connected allocation for the original instance if and only if there is an exact cover. Indeed if the allocation π when focussing on the non-copied players only is a perfect connected allocation for the original instance, then the argument in the proof of Theorem 3 tells us an exact cover.

Conversely, suppose that there is an exact cover of size r. Then, we have seen in the proof of Theorem 3 that there is a a perfect connected allocation π^* for the original instance. Now assume towards a contradiction that the allocation π restricted to the non-copied players is not a perfect connected allocation for the original instance; then, one can construct a Pareto-improvement of π as follows.

- Each copied player $\hat{i}$ receives the bundle $\pi(\hat{i})$ except that we remove any non-copied items from this bundle; and
- Each non-copied player i receives the bundle $\pi^*(i)$.

The resulting allocation is a Pareto-improvement of π, since at least one of non-copied players is strictly better off and the copied players obtain the same utility in π and π^*. This contradicts the Pareto-optimality of π.

An analogous argument holds when the player receiving the center is a non-copied player. Hence, it follows that, by looking at the bundle of each player in a Pareto-optimal outcome, one can decide the existence of an exact cover. □

We note that the graph constructed in the above proof has pathwidth 3 and diameter 7, so hardness holds even for trees of bounded pathwidth and bounded diameter. An interesting open question is whether hardness holds for graphs with bounded maximum degree.

4 Pareto-Optimality & EF1 on Paths

In the previous section, we were aiming to find any Pareto-optimum, and obtained a positive result for the important case where G is a path. Now we aim higher, wanting to find an efficient allocation which is also fair.

When there are no connectivity requirements, it is known that efficiency and fairness are compatible: Caragiannis et al. [12] showed that an allocation maximizing the *Nash product* of agents' utilities is both Pareto-optimal and EF1. While it is NP-hard to compute the Nash solution, Barman et al. [6] designed a (pseudo-)polynomial time algorithm which finds an allocation satisfying these two properties.

In our model, unfortunately, EF1 is incompatible with Pareto-optimality, even when G is a path. The following construction only requires binary utilities.

Example 1. Consider an instance of four players a_1, a_2, a_3, b and a path depicted in Figure 2. Player b approves only black vertices whereas the other three players

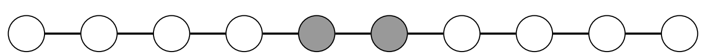

Fig. 2. Example where no allocation satisfies both PO and EF1.

approve only white vertices. In an EF1 allocation, all the black vertices cannot be allocated to one of the latter players, since otherwise the envy of player b would not be eliminated by deleting one item; hence, players a_1, a_2, and a_3 cannot receive a bundle containing all the black vertices. Further in a Pareto-optimal allocation, player b cannot be assigned to a white vertex because otherwise giving it to others strictly improves the utility of that player without decreasing the utility of b. Thus, if a Pareto-optimal EF1 allocation existed, one of a_1, a_2, and a_3 would obtain four white vertices and the other players would obtain only two white vertices, violating the EF1 requirement.

□

The following alternative example shows that Pareto-optimality and EF1 conflict in an even more restricted setting, where each agent's approval set is an interval.

Example 2. Consider an instance of three players a_1, a_2, and b, and a path depicted in Figure 3, where player b approves only black vertices and the other three players a_1 and a_2 approve every vertex. Again, in any EF1 allocation, all

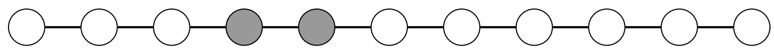

Fig. 3. Example where no allocation satisfies PO and EF1 even when each player approves an interval.

the black vertices cannot be allocated to one of the latter players, since otherwise the envy of player b would not be eliminated by deleting one item. Also, in any Pareto-optimal allocation, player b cannot receive a white vertex because otherwise giving it to others strictly improves the utility of that player without decreasing the utility of b. However, if a_1 and a_2 do not obtain the first three vertices, reallocating the first three vertices to one of them and the last six vertices to the other is a Pareto-improvement. If one of a_1 and a_2 obtains the first three vertices, then the other receives at least the last six white vertices by Pareto-optimality; thus, the envy towards the player receiving six white vertices is not bounded up to one good. In either case, there is no Pareto-optimal and EF1 allocation. □

Given that we do not have an existence guarantee, a natural question is whether it is easy to decide whether a given instance admits a Pareto-optimal allocation satisfying EF1. Using the above example, we prove that the problem is NP-hard. Note that the obvious complexity upper bound is Σ_2^p; an interesting open problem is whether the problem is complete for this class.

Theorem 5. *It is* NP*-hard to decide whether a Pareto-optimal EF1 connected allocation exists when G is a path, even if utilities are binary.*

Proof. Given an instance $(X, \mathcal{S})$ of X3C, we create the same instance of CFD as in the proof of Theorem 3. We create additional items and players as follows; unless we explicitly specify, each player has zero utility for the additional items.

	○	○	○	○	●	●	○	○	○	○
i_S, a_S^1, a_S^2 :	1	1	1	1	0	0	1	1	1	1
b_S :	0	0	0	0	1	1	0	0	0	0

Table 2. Empty instance I_S

	○	○	○	●	●	○	○	○	○	○
a_x^1, a_x^2 :	1	1	1	1	1	1	1	1	1	1
i_x :	0	0	0	1	1	0	0	0	0	0

Table 3. Empty instance I_x

I_S gadget: We create an empty PO and EF1 instance I_S for each player i_S $(S \in \mathcal{S})$. The empty instance I_S consists of a path of two black vertices and eight

white vertices as depicted in Figure 2, together with player i_S, players a_S^1 and a_S^2, and player b_S. Each i_S, a_S^1, and a_S^2 approve the white vertices of the associated path while b_S approves the corresponding black vertices only. See Table 2 for these valuations.

I_x gadget: We create an empty PO and EF1 instance I_x for each player i_x ($x \in X$). The empty instance I_x consists of a path of two black vertices and nine white vertices as in Figure 3, together with player i_x and players a_x^1 and a_x^2. Each i_x approves the black vertices of the associated path whereas players a_x^1 and a_x^2 approve every vertex on the path. See Table 3 for these valuations.

Dummies: We create a dummy player for each connected component of the graph constructed so far. We connect the connected components by constructing paths of two dummy vertices. Each dummy player only approves the dummy vertices on the path right next to their associated connected component. Then in any EF1 allocation, none of the non-dummy player obtains a bundle containing both of the dummy vertices.

It can be verified that there is a Pareto-optimal EF1 connected allocation if and only if there is an exact cover; the proof can be found in the appendix. □

5 Pareto-Optimality & MMS on Paths

In the previous section, we saw that deciding the existence of an allocation that is efficient and satisfies EF1 is computationally hard, even if G is a path. Part of the reason is that envy-freeness notions and Pareto-optimality are not natural companions: it is easy to construct examples where some allocation is envy-free, yet by Pareto-improving the allocation, we introduce envy.

An alternative notion of fairness avoids this problem: Pareto-improving upon an MMS allocation preserves the MMS property. Bouveret et al. [9] showed that if G is a tree, then an MMS allocation always exists (and can be found efficiently). An implication of their result is that on trees, there is an allocation that is both fair and efficient: start with some MMS allocation, and repeatedly construct Pareto-improvements until we find a Pareto-optimal allocation, which must still satisfy the MMS property.

While existence is guaranteed, it is unclear whether we can find such an allocation efficiently. Certainly, by the negative results of Section 3.2, this is not possible when G is an arbitrary tree. What about the case when G is a path? The answer is again negative: a Pareto-optimal MMS allocation cannot be found efficiently, even on a path.

Theorem 6. *Unless* P = NP*, there is no polynomial time algorithm which finds a Pareto-optimal MMS connected allocation when G is a path, even if utilities are binary.*

Proof. Again, given an instance $(X, \mathcal{S})$ of X3C, we create the same instance of CFD as in the proof of Theorem 3. For each connected component, we introduce a dummy player. Suppose our instance now has n players in total. We create a path by connecting the connected components by paths of length n. Each dummy player only approves each good on the path to the right of their associated component and has zero utility for the other goods. Note that dummy players have positive MMS value, since it is possible to partition the items into n connected bundles so that each contains an item approved by the dummy player. Thus, in any MMS allocation, the dummy player must obtain one of the items in the connecting path. The original players have zero utility for the dummy items.

Let π be a Pareto-optimal MMS connected allocation. Then no bundle contains more than one connected component of the original graph, since otherwise the MMS requirement of a dummy player would be violated. Hence the original argument works as applied to the original players. Specifically, when restricted to the non-dummy players, π is a perfect allocation of the original instance if and only if there is an exact cover. Thus, by looking at a Pareto-optimal MMS connected allocation, one can efficiently decide the existence of an exact cover. $\square$

For $\alpha \in (0, 1]$, we say that an allocation π is α-MMS if $u_i(\pi(i)) \geqslant \alpha \cdot \mathrm{mms}_i(I)$ for all $i \in N$. The above proof implies that we cannot in polynomial time find a Pareto-optimal allocation that is α-MMS, for any $\alpha > 0$.

6 Conclusion

In this work, we have taken the first steps in studying a connection between underlying graph structure and the ability to guarantee efficiency in fair division. We showed that although finding a Pareto-optimal allocation is easy for paths and stars, this does not extend to general trees. Further, we proved that with the additional fairness requirements, the corresponding solution becomes NP-hard even when the underlying item graph is a path. We have also seen that a Pareto-optimal and EF1 allocation may not exist with the contiguity requirement while the standard fair division instance always admits such an allocation.

While we have focused on the divisions of goods, studying an allocation of *chores* with graph-connectivity constraints would be particularly interesting; specifically, one may ask what graph structure ensures tractability results in terms of both existence and complexity guarantees. Finally, we note that several very recent papers studied the fair division problem over social networks [1, 8, 13, 10] where their graph describes the envy relation between agents. A particular focus is laid on *local envy-freeness*, requiring that each player does not envy the bundles of her neighbours. Although our graph describes a relationship among *items* and hence is different from theirs, it would be interesting to analyze 'intermediate' cases; that is, suppose that we only focus on the envy between a pair of agents who are allocated adjacent bundles, what graph structure guarantees local envy-freeness and optimal efficiency?

References

1. Abebe, R., Kleinberg, J.M., Parkes, D.C.: Fair division via social comparison. In: Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). pp. 281–289 (2017)

2. Aumann, Y., Dombb, Y.: The efficiency of fair division with connected pieces. ACM Transactions on Economics and Computation **3**, 23:1–23:16 (2015)
3. Aumann, Y., Dombb, Y., Hassidim, A.: Computing socially-efficient cake divisions. In: Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). pp. 343–350 (2013)
4. Aziz, H., Biró, P., Lang, J., Lesca, J., Monnot, J.: Optimal reallocation under additive and ordinal preferences. In: Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). pp. 402–410 (2016)
5. Aziz, H., Brandt, F., Harrenstein, P.: Pareto optimality in coalition formation. Games and Economic Behavior **82**, 562–581 (2013)
6. Barman, S., Murthy, S.K.K., Vaish, R.: Finding fair and efficient allocations. arXiv:1707.04731 [cs.GT] (2017), http://arxiv.org/abs/1707.04731
7. Bei, X., Chen, N., Hua, X., Tao, B., Yang, E.: Optimal proportional cake cutting with connected pieces. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI). pp. 1263–1269 (2012)
8. Bei, X., Qiao, Y., Zhang, S.: Networked fairness in cake cutting. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI). pp. 3632–3638 (2017)
9. Bouveret, S., Cechlárová, K., Elkind, E., Igarashi, A., Peters, D.: Fair division of a graph. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI). pp. 135 – 141 (2017)
10. Bredereck, R., Kaczmarczyk, A., Niedermeier, R.: Envy-free allocations respecting social networks. In: Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2018)
11. Budish, E.: The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. Journal of Political Economy **119**(6), 1061–1103 (2011)
12. Caragiannis, I., Kurokawa, D., Moulin, H., Procaccia, A.D., Shah, N., Wang, J.: The Unreasonable Fairness of Maximum Nash Welfare. In: Proceedings of the 17th ACM Conference on Economics and Computation (ACM-EC). pp. 305–322 (2016)
13. Gourvés, L., Lesca, J., Wilczynski, A.: Object allocation via swaps along a social network. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI). pp. 213–219 (2017)
14. Korte, B., Vygen, J.: Combinatorial Optimization - Theory and Algorithms. Springer-Verlag Berlin Heidelberg (2006)
15. Lindner, C., Rothe, J.: Cake-Cutting: Fair Division of Divisible Goods, pp. 395–491. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
16. Lipton, R.J., Markakis, E., Mossel, E., Saberi, A.: On approximately fair allocations of indivisible goods. In: Proceedings of the 5th ACM Conference on Electronic Commerce (ACM-EC). pp. 125–131. ACM Press (2004)
17. Stromquist, W.: How to cut a cake fairly. American Mathematical Monthly **87**, 640–644 (1980)
18. Stromquist, W.: Envy-free divisions cannot be found by finite protocols. The Electronic Journal of Combinatorics **15**, 145–158 (2008)
19. Suksompong, W.: Fairly allocating contiguous blocks of indivisible items. In: Proceedings of the 10th International Symposium on Algorithmic Game Theory (SAGT). pp. 333–344. Lecture Notes in Computer Science (2017), extended version available as arXiv:1707.00345 [cs.GT].

Appendix

Proof of Theorem 5

Proof. Suppose that there is a Pareto-optimal EF1 connected allocation π. By EF1, each non-dummy player cannot obtain more than one connected component of the original graph. Further, for each $S \in \mathcal{S}$, none of the players a_S^1, a_S^2, and b_S approves the vertices outside of I_S. Thus, in any Pareto-optimal and EF1 allocation,

- each player b_S must be allocated two black vertices of the associated path;
- each of the agents a_S^1 and a_S^2 must be allocated the four white vertices at the end; and
- each player i_S receives a bundle outside of I_S that contains at least three approved items.

Otherwise there would be no Pareto-optimal and EF1 allocation as we have seen in Example 1. Similarly, for each $x \in X$, none of the players except for i_x approve the vertices outside I_x. Hence, in any Pareto-optimal and EF1 allocation,

- one of the agents a_x^1 and a_x^2 obtain two black vertices of the path in I_x; and
- each player i_x receives a bundle outside of I_x that contains at least one approved item.

Thus, when focusing on the x-players i_x and S-players i_S, the allocation π is a perfect allocation of the original instance; hence there is an exact cover as we have seen in the proof of Theorem 3.

Conversely, suppose that there is an exact cover. Then, as we proved in the proof of Theorem 3, there is a perfect allocation of the original instance where each player i_x receives one approved item and each player i_S receives three approved items. We extend this allocation as follows

- each dummy player obtain the associated dummy vertices;
- for each $S \in \mathcal{S}$, a_S^1 receives the first four white vertices of the path in I_S, a_S^2 receives the last four white vertices, and b_S obtains the two black vertices; and
- for each $x \in X$, a_x^1 the first five vertices of the path in I_x, and a_x^2 obtains the last six vertices.

The resulting allocation is Pareto-optimal since each item is allocated to a player who approves it; also, it can be easily seen to satisfy EF1. $\square$