Fair Division of a Graph

Sylvain Bouveret
LIG - Grenoble INP, France
sylvain.bouveret@imag.fr

Katarína Cechlárová
P.J. Šafárik University, Slovakia
katarina.cechlarova@upjs.sk

Edith Elkind
University of Oxford, UK
elkind@cs.ox.ac.uk

Ayumi Igarashi
University of Oxford, UK
ayumi.igarashi@cs.ox.ac.uk

Dominik Peters
University of Oxford, UK
dominik.peters@cs.ox.ac.uk

Abstract

We consider fair allocation of indivisible items under an additional 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. This framework captures, e.g., fair allocation of land plots, where the graph describes the accessibility relation among the plots. We focus on agents that have additive utilities for the items, and consider several common fair division solution concepts, such as proportionality, envy-freeness and maximin share guarantee. While finding good allocations according to these solution concepts is computationally hard in general, we design efficient algorithms for special cases where the underlying graph has simple structure, and/or the number of agents—or, less restrictively, the number of agent types—is small. In particular, despite non-existence results in the general case, we prove that for acyclic graphs a maximin share allocation always exists and can be found efficiently.

1 Introduction

The department of computer science at University X is about to move to a new building. Each research group has preferences over rooms, but it would also be desirable for each group to have a contiguous set of offices, to facilitate communication. This situation can be seen as a problem of fair division (where agents are research groups and items are offices) with an additional connectivity requirement. This constraint could be captured by an undirected graph whose vertices are rooms (items) and there is an edge between two vertices if the respective rooms are adjacent; each agent should obtain a connected piece of this graph.

In this paper, we introduce and study a formal model for such scenarios. Specifically, we consider the problem of fair allocation of indivisible items in settings where there is a graph capturing the dependency relation between items, and each agent's share has to be connected in this graph. Besides the example in the first paragraph, our model captures a variety of applications, such as time-sharing a processor where tasks can be switched only at pre-defined times, allocating a set of indivisible land plots, or assigning administrative duties to members of an academic department, where there are dependencies among tasks (e.g., dealing with incoming foreign students has some overlap with preparing study programmes in foreign languages, but not with fire safety).

Our contribution We propose a framework for fair division under connectivity constraints, and investigate the complexity of finding good allocations in this framework according to three well-studied solution concepts: proportionality, envy-freeness (in conjunction with completeness), and maximin share guarantee. We focus on additive utility functions.

For proportionality and envy-freeness, we obtain hardness results even for very simple graphs: finding proportional allocations turns out to be NP-hard even for paths, and finding complete envy-free allocations is NP-hard both for paths and for stars. Nevertheless, we also obtain some positive results for these solution concepts. In particular, both proportional and complete envy-free allocations can be found efficiently when the graph is a path and agents can be classified into a small number of *types*, where agents are said to have the same type when they have the same preferences over items.[1] If we assume that not just the number of player types, but the actual number of players is small, we obtain an efficient algorithm for finding proportional allocations on arbitrary trees.

Recently, several papers have studied the concept of the *maximin share guarantee* (MMS) [Budish, 2011], which captures a desirable property of allocations that is easy to achieve for divisible items via cut-and-choose protocols. For indivisible goods, such allocations need not exist [Procaccia and Wang, 2014; Kurokawa *et al.*, 2016]. We prove a strong positive result for our setting: an MMS allocation always exists if the underlying graph is a tree, and can be computed efficiently. Our algorithm is an adaptation of the classic last-diminisher procedure for the divisible case. In contrast, we provide an example where the underlying graph is a cycle of length 8 and there is no MMS allocation. We believe that these results are useful for developing an intuitive understanding of the concept of MMS; in particular, our example for the cycle is much simpler than known examples of instances with no MMS allocation in the absence of graph constraints.

---

[1] The same parameter was used by Brânzei *et al.* [2016] to obtain results for maximizing social welfare; similar ideas have been used in the context of coalition formation [Shrot *et al.*, 2010; Aziz and De Keijzer, 2011].

Related work Fair allocation of indivisible items has received a considerable amount of attention in the (computational) social choice literature; we refer the reader to a survey by [Bouveret *et al.*, 2016]. However, ours is the first attempt to impose a graph-based constraint on players' bundles. In contrast, in the context of fair allocation of *divisible* items (also known as cake-cutting) contiguity is a well-studied requirement. For instance, [Stromquist, 1980] showed that an envy-free division in which each player receives a single contiguous piece always exists, but it cannot be obtained by a finite algorithm, even for three players [Stromquist, 2008]. These results extend to equitable division with contiguous pieces [Cechlárová *et al.*, 2013; Aumann and Dombb, 2015; Cechlárová and Pillárová, 2012]. [Bei *et al.*, 2012] consider fair allocations with contiguous pieces that approximately maximize social welfare; [Aumann *et al.*, 2013] investigate a variant of this question without fairness constraints.

[Conitzer *et al.*, 2004] analyze a combinatorial auction setting that is somewhat similar to ours: in their model, too, there is an undirected graph describing connections between items, and each agent's bid is connected with respect to this graph. They provide an algorithm for finding an allocation that maximizes the social welfare and is in FPT with respect to the treewidth of the item graph. [Aumann *et al.*, 2015] consider auctioning of a time interval, and obtain results both for the case of pre-determined time slots (which corresponds to the model of [Conitzer *et al.*, 2004], with the item graph being a line) and for the case where the interval can be cut into arbitrary slots (which is similar in spirit to cake-cutting). However, neither paper considers any fairness constraints.

Two very recent papers, like ours, combine graphs and fair division. [Chevaleyre *et al.*, 2017] consider the setting where agents are located in vertices of a graph. Each agent has an initial endowment of goods and can trade with her neighbors in the graph. The authors ask what outcomes can be achieved by a sequence of mutually beneficial deals. In the work of [Abebe *et al.*, 2017], the graph describes a visibility relation: agents are located in vertices and an agent can only envy agents who are adjacent to her. In contrast, in our model graphs represent the relationship between items rather than agents.

2 Our Model

We study fair allocation of indivisible goods where each allocated bundle is connected in an underlying graph.

Definition 2.1. An instance of the *connected fair division problem (CFD)* is a triple $I = (G, N, \mathcal{U})$ where

- $G = (V, E)$ is an undirected graph,
- $N = \{1, \ldots, n\}$ is a set of *players*, or *agents*,
- $\mathcal{U}$ is an n-tuple of utility functions $u_i : V \to \mathbb{R}_{\geqslant 0}$, where $\sum_{v \in V} u_i(v) = 1$ for each $i \in N$.

We refer to elements of V as *items*, and denote the number of items by m.

Note that when G is a clique, CFD is equivalent to the classic problem of fair allocation with indivisible items.

For each $X \subseteq V$, we set $u_i(X) = \sum_{v \in X} u_i(v)$, so valuations in this paper are always additive. Two players $i, j \in N$

are of the *same type* if $u_i(v) = u_j(v)$ for all $v \in V$. We denote the number of player types in a given instance by p.

An *allocation* is a function $\pi : N \to 2^V$ assigning each player a bundle of items. An allocation π is *valid* if for each player $i \in N$ the bundle $\pi(i)$ is connected in G and no item is allocated twice, so that $\pi(i) \cap \pi(j) = \emptyset$ for each pair of distinct players $i, j \in N$. We say that a valid allocation π is

- *proportional* if $u_i(\pi(i)) \geqslant \frac{1}{n}$ for all $i \in N$,
- *envy-free* if $u_i(\pi(i)) \geqslant u_i(\pi(j))$ for all $i, j \in N$, and
- *complete* if $\bigcup_{i \in N} \pi(i) = V$.

Notice that an allocation that gives everybody an empty bundle is envy-free, so, to better express the idea of fairness, the requirement of envy-freeness is typically accompanied by completeness or Pareto-optimality.

We also consider *maximin share (MMS) allocations* [Budish, 2011], adapting the usual definition to our 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

$$\text{mms}_i(I) = \max_{(P_1, \ldots, P_n) \in \Pi_n} \min_{j \in \{1, \ldots, n\}} u_i(P_j).$$

Note that since we are only taking the maximum over connected partitions, these values may be lower than in the general setting without graph constraints. A valid allocation π is a *maximin share (MMS) allocation* if we have $u_i(\pi(i)) \geqslant \text{mms}_i(I)$ for each player $i \in N$.

We consider the following computational problems that all take an instance $I = (G, N, \mathcal{U})$ of the connected fair division problem as input. For computational purposes, we assume that utility functions take values in $\mathbb{Q}$. Hardness results will use unary encodings of utility values (unless noted otherwise).

- PROP-CFD: Does I admit a proportional valid allocation?
- COMPLETE-EF-CFD: Does I admit a complete envy-free valid allocation?
- MMS-CFD: Does I admit an MMS allocation?

We note that, given a valid allocation, one can check in polynomial time whether it is proportional or envy-free; thus, the respective computational problems are in NP.

In what follows, we assume that the number of items m is at least as large as the number of players n, since otherwise at least one player gets nothing. Also, given a positive integer k, we write $[k]$ to denote the set $\{1, \ldots, k\}$.

3 Proportionality

We start with the bad news: it is hard to find a proportional allocation, even if the graph G is a path.

Theorem 3.1. PROP-CFD *is* NP-*complete even if* G *is a path.*

Proof. We describe a polynomial-time reduction from the NP-complete problem EXACT-3-COVER (X3C) [Garey and Johnson, 1979]. Recall that an instance of X3C is given by a set of elements $X = \{x_1, x_2, \ldots, x_{3s}\}$ and a family $\mathcal{T} = \{T_1, T_2, \ldots, T_r\}$ of three-element subsets of X; it is a 'yes'-instance if and only if X can be covered by s sets from $\mathcal{T}$.
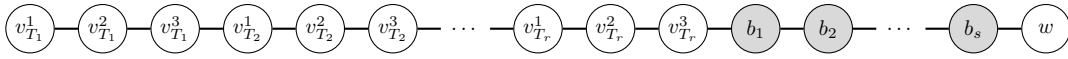
Figure 1: Graph constructed in the proof of Theorem 3.1.

This problem remains NP-complete if for each element $x \in X$ its frequency $p_x = |\{T \in \mathcal{T} : x \in T\}|$ is at most 3.

Consider an instance $J = (X, \mathcal{T})$ of X3C; for each $T \in \mathcal{T}$, we denote the elements of T by x_T^1, x_T^2, x_T^3. We construct an instance I of PROP-CFD as follows. There are three small vertices v_T^1, v_T^2, v_T^3 for each set $T \in \mathcal{T}$, a set of s big vertices $B = \{b_1, b_2, \ldots, b_s\}$ and a dummy vertex w. The edges of G are shown in Figure 1.

There is one player i_T for each $T \in \mathcal{T}$, one player i_x for each $x \in X$ and one dummy player d. Hence the total number of players is $n = 3s + r + 1$. Define the utilities as:

$$u_{i_T}(v) = \begin{cases} 1/(3n) & \text{if } v = v_T^k \\ 1/n & \text{if } v \in B \\ (n-s-1)/n & \text{if } v = w \\ 0 & \text{otherwise} \end{cases}$$

$$u_{i_x}(v) = \begin{cases} 1/n & \text{if } v = v_T^k \text{ and } x \in T \\ (n-3p_x)/n & \text{if } v = w \\ 0 & \text{otherwise} \end{cases}$$

$$u_d(v) = \begin{cases} 1 & \text{if } v = w \\ 0 & \text{otherwise} \end{cases}$$

By construction $u_i(V) = 1$ for each $i \in N$. As player d assigns a positive value to vertex w only, she must receive this vertex in every proportional allocation. Given that w is allocated to d, an allocation is proportional if and only if each player i_x receives a small vertex v_T^k such that $x \in T$, and each player i_T receives vertices v_T^1, v_T^2, v_T^3 (a triple interval) or a vertex from B.

Suppose that J admits a cover $\mathcal{T}'$ of size s. Let μ be a matching between $\mathcal{T}'$ and B. Assign intervals to players i_x and i_T as follows:

- For each $T \in \mathcal{T}'$, player i_T is assigned to vertex $\mu(T)$.
- For each $T \notin \mathcal{T}'$, player i_T is assigned to the triple interval v_T^1, v_T^2, v_T^3.
- Each player i_x is assigned to the small vertex v_T^k such that $x = x_T^k$ and $T \in \mathcal{T}'$.

Then each player is assigned one connected piece and her value for that piece is at least $1/n$.

Conversely, suppose that I admits a proportional valid allocation. As $|B| = s$, the number of T-players assigned to triple intervals is $r - s$. Hence, the number of triple intervals available for x-players is s, and the respective sets constitute an exact cover for X. □

In contrast, if G is a star, finding a proportional allocation is easy. Our algorithm for this problem, as well as all other algorithms in this section, use matching techniques and can be adapted to also find valid allocations that maximize egalitarian welfare, i.e., the utility of the worst-off agent.

Theorem 3.2. PROP-CFD *is solvable in polynomial time if G is a star.*

3.1 Bounded Number of Agent Types

If the underlying graph is a path and all players are of the same type, then a simple greedy algorithm finds a proportional allocation (or reports that none exists) in linear time: we build connected pieces one by one, by moving along the path from left to right and adding vertices to the current piece until its value to a player reaches $1/n$; at this point we start building a new piece. This procedure creates at most n pieces; a proportional valid allocation exists if and only if it creates exactly n pieces. More generally, if G is a path and the number of agent types is bounded by a constant, a simple dynamic program can check the existence of a proportional allocation in polynomial time. A problem is *slice-wise polynomial* (XP) with respect to a parameter k if each instance I of this problem can be solved in time $|I|^{f(k)}$ where f is a computable function.

Theorem 3.3. PROP-CFD *is in XP with respect to the number of player types p if G is a path.*

Proof. Let $G = (V, E)$, where $V = \{v_1, \ldots, v_m\}$, $E = \{\{v_i, v_{i+1}\} : i \in [m-1]\}$. Suppose there are n_t players of type t, for $t \in [p]$. We say that a player is *happy* if she gets a connected piece of value at least $1/n$. Let $V_0 = \emptyset$ and $V_i = \{v_1, \ldots, v_i\}, i > 1$.

For $i = 0, \ldots, m$, and a collection of indices $j_1, \ldots, j_p$ such that $0 \leqslant j_k \leqslant n$ for each $k \in [p]$, let $A_i[j_1, \ldots, j_p] = 1$ if there exists a valid partial allocation π of V_i with j_k happy agents of type k, $k \in [p]$, and let $A_i[j_1, \ldots, j_p] = 0$ otherwise. Clearly, $A_0[j_1, \ldots, j_p] = 1$ if and only if $j_k = 0$ for all $k \in [p]$. For $i = 1, \ldots, m$, we have $A_i[j_1, \ldots, j_p] = 1$ if and only if there exists a value $s < i$ and $t \in [p]$ such that $A_s[j_1, \ldots, j_t - 1, \ldots, j_p] = 1$ and a player of type t values the set of items $\{v_{s+1}, \ldots, v_i\}$ at $1/n$ or higher.

A proportional allocation exists if $A_m[j_1, \ldots, j_p] = 1$ for some collection of indices $j_1, \ldots, j_p$ such that $j_t \geqslant n_t$ for all $t \in [p]$. There are at most $(m+1)(n+1)^p$ values to compute, and each value can be found in time $O(mt)$ using unit cost arithmetics. Thus, PROP-CFD is in XP with respect to p. □

3.2 Bounded Number of Agents

Theorem 3.3 immediately implies that PROP-CFD can be solved in polynomial time if the number of agents n is bounded by a constant and the graph is a path. A more careful argument shows that PROP-CFD on trees, is, in fact, fixed parameter tractable with respect to this (weaker) parameter. A problem is *fixed parameter tractable* (FPT) with respect to a parameter k if each instance I of this problem can be solved in time $f(k)\text{poly}(|I|)$ where f is a function that depends only on k.

Theorem 3.4. PROP-CFD *is in FPT with respect to n when G is a tree.*

Proof sketch. We turn G into a rooted tree by choosing an arbitrary node r as the root. Given a vertex v, we denote by $C(v)$ the set of children of v and by $D(v)$ the set of descendants of v (including v) in the rooted tree.

For each $v \in V$, $S \subsetneq N$, and $i \in N \setminus S$, let $\Pi_{i,v,S}$ be the set of all valid allocations $\pi : S \cup \{i\} \to 2^{D(v)}$ with $v \in \pi(i)$ and $u_j(\pi(j)) \geqslant 1/n$ for all $j \in S$, and define

$$A_v[i, S] = \max\{u_i(\pi(i)) : \pi \in \Pi_{i,v,S}\};$$

by convention, $A_v[i, S] = -\infty$ when $\Pi_{i,v,S}$ is empty. Note that we have a 'yes'-instance of PROP-CFD if and only if $A_r[i, N \setminus \{i\}] \geqslant 1/n$ for some $i \in N$.

To complete the proof, we show that the values $A_v[i, S]$ can be efficiently computed in a bottom-up manner. We observe that for every internal vertex v of the rooted tree, in each allocation $\pi \in \Pi_{i,v,S}$ the bundle of each player in S is fully contained in a subtree $D(z)$ for some $z \in C(v)$. This induces a partition of players in S into $|C(v)|$ groups. We can go through all possible partitions of S with at most $|C(v)|$ parts and try to find a valid allocation compatible with this partition, maximizing the utility of each agent. Such an allocation can be found efficiently by solving a matching problem. We omit the details due to space constraints. $\square$

We note that placing strong constraints on the underlying graph is crucial for obtaining the easiness results in Theorems 3.3 and 3.4. This is illustrated by the following simple proposition, obtained by adapting a proof by [Demko and Hill, 1998] for the standard setting (with no graph constraints), which shows that the XP membership with respect to the number of players/types cannot be extended to arbitrary graphs.

Proposition 3.5. *When utilities are encoded in binary, PROP-CFD is NP-complete even for $n = 2$, $p = 1$, and even if the underlying graph G is bipartite.*

4 Envy-freeness

Envy-freeness turns out to be computationally more challenging than proportionality: finding a complete envy-free allocation is NP-hard even if the underlying graph is a star (for complete graphs, this result is shown by Lipton *et al.* [2004]).

Theorem 4.1. COMPLETE-EF-CFD *is NP-complete even if G is a star.*

Proof. Our hardness proof proceeds by a reduction from IN-DEPENDENT SET. Recall that an instance of INDEPENDENT SET is given by an undirected graph (W, L) and an integer k; it is a 'yes'-instance if and only if (W, L) contains an independent set of size k. Given an instance (W, L) of INDEPENDENT SET, we construct an instance of COMPLETE-EF-CFD as follows. For each vertex $w \in W$ we create an item w and a player i_w. Similarly, for each edge $\ell \in L$ we create an item ℓ and a player i_ℓ. We also create a set of dummy items D with $|D| = k$, as well as an item c and a player i_c. The graph G is a star with center c and set of leaves $W \cup L \cup D$. Finally, define utility functions as follows.

- For each $w \in W$, we set $u_{i_w}(w) = 1/(k+1)$ and $u_{i_w}(d) = 1/(k+1)$ for each $d \in D$.
- For each $\ell \in L$ with $\ell = \{x, y\}$, we set $u_{i_\ell}(\ell) = 3/7$, $u_{i_\ell}(x) = u_{i_\ell}(y) = 2/7$.
- We set $u_{i_c}(c) = 1$.
- All other utilities are set to 0.

We will now argue that there exists an independent set of size k in the graph (W, L) if and only if this instance of CFD admits a complete envy-free valid allocation.

Suppose there exists an independent set $X \subseteq W$ of size k. We construct an allocation π as follows:

- player i_c receives $X \cup \{c\}$;
- for $w \in W \setminus X$, player i_w receives w;
- for $w \in X$, player i_w receives one item in D;
- for $\ell \in L$, player i_ℓ receives ℓ.

Clearly, π is a complete valid allocation. It remains to show that π is envy-free. First, player i_c does not envy any other player since she receives all her positive-utility items. Vertex players $\{i_w : w \in W\}$ receive utility $1/(k + 1)$ in π; they could only envy someone who has multiple dummies, but no one does. Edge players $\{i_\ell : \ell \in L\}$ receive utility $3/7$ in π; the only way an edge player i_ℓ could envy another player is if that player got both items corresponding to endpoints of ℓ. But the only player who receives more than one vertex item is player i_c whose items correspond to an independent set. So no player envies anyone, and π is envy-free.

Conversely, suppose that there is a complete envy-free valid allocation π. By completeness, π allocates the central piece c to some player. If i_c does not receive c then she would envy the player who receives it; so $c \in \pi(i_c)$. Thus, every other player receives at most one item. Since π is complete, this means that i_c gets at least k leaf items. Further, if i_ℓ does not receive ℓ, she would envy the player who receives it, so $\pi(i_\ell) = \{\ell\}$. Next, consider the bundle of player i_c. If it contains more than one dummy item, vertex players would envy i_c. Thus, it contains at least one item $w \in W$. If $\pi(i_c)$ also contains a dummy item, i_w would envy i_c, so $\pi(i_c)$ consists of c and k vertex items. Now, if there is an edge $\ell = (x, y)$ such that $x, y \in \pi(i_c)$, then player i_ℓ envies i_c. Hence, $\pi(i_c) \setminus \{c\}$ forms an independent set of size k in (W, L). $\square$

We also obtain a hardness result for paths; the proof is similar to that of Theorem 3.1.

Theorem 4.2. *The problem* COMPLETE-EF-CFD *is NP-complete even if G is a path.*

On the positive side, just as for PROP-CFD, the problem COMPLETE-EF-CFD is also in XP with respect to the number of player types p, as long as G is a path.

Theorem 4.3. COMPLETE-EF-CFD *is in XP with respect to the number of player types p if G is a path.*

Proof sketch. Note that for an allocation to be envy-free, all pieces assigned to players of a given type should have the same value to players of that type. When G is a path, there are only $\binom{m+1}{2} \leqslant m^2$ different connected bundles. Hence there are at most m^2 many possibilities for the utility that a player of a given type can obtain in a valid allocation.

Our algorithm works as follows. For each player type, it guesses the utility that players of that type assign to their pieces (this guessing can be implemented by going over all possibilities, as there are at most $(m^2)^p$ of them). It then proceeds similarly to the dynamic programming algorithm in the proof of Theorem 3.3; the only difference is that, when

creating a piece of the form $\{v_{s+1}, \ldots, v_i\}$ for a player of a given type, it checks that the utility of that player type for this piece is what it guessed for that type, and that other players' utility for this piece is at most their guessed utility. □

5 Maximin Share Guarantee

After Budish [2011] introduced the notion of an MMS allocation, it was open for some time whether every allocation problem (without connectivity constraints) admitted such an allocation. Procaccia and Wang [2014] found a counterexample. A family of more compact examples was found by Kurokawa *et al.* [2016]; these examples implicitly use an underlying grid graph; hence, for grid graphs, existence of MMS allocations is not guaranteed. Here, we show that for *trees* an MMS allocation always exists. Our argument is constructive, and our algorithm corresponds to a discrete version of the last-diminisher method, which ensures proportionality while cutting a divisible resource [see, e.g., Brams and Taylor, 1996]. This method proceeds by letting one player identify a bundle of items. Every other player, in order, then has the option to *diminish* this bundle by removing some of the items from it. The last player who chose to diminish is allocated the (diminished) bundle. The same procedure is then applied to divide the rest of the cake among the remaining $n-1$ players.

We first describe an efficient procedure that guarantees each player a pre-specified level of utility.

Proposition 5.1. *Let $I = (G, N, \mathcal{U})$ be an instance of CFD where G is a tree and let $(q_i)_{i \in N}$ be an n-tuple of rational numbers. If $\mathrm{mms}_i(I) \geqslant q_i$ for all $i \in N$, then there exists a valid allocation π such that each player $i \in N$ receives the bundle of value at least q_i, i.e., $u_i(\pi(i)) \geqslant q_i$. Moreover, one can compute such an allocation in polynomial time.*

Proof. We will give an informal description of our recursive algorithm $\mathcal{A}$ (Algorithm 1), followed by pseudocode. For each $X \subseteq V$, we let $G \setminus X$ denote the subgraph induced by $V \setminus X$; also, we denote the restriction of u_i to X by $u_i|_X$.

The algorithm first checks whether its input graph G' has a value of at least q_i for each player $i \in N'$; if this is not the case, it fails. Then, if there is only one player, the algorithm simply returns the allocation that assigns all items to that player. When there are at least two players, $\mathcal{A}$ turns the graph into a rooted tree by choosing an arbitrary node as its root; denote by $D(v)$ the set of descendants of a vertex v in this rooted tree. Then each player i finds a vertex v_i such that his value for $D(v_i)$ is at least q_i, but for each child w of v his value for $D(w)$ is less than q_i. The algorithm then allocates $D(v_i)$ to the *last-diminisher* i whose vertex v_i has minimal height. The player i exits with the bundle $D(v_i)$, and the same algorithm $\mathcal{A}$ is called on the remaining instance (see Fig. 2).

It is immediate that $\mathcal{A}$ runs in polynomial time. Let $I_n, \ldots, I_1$ be the sequence of instances constructed by $\mathcal{A}$ when called on I and $(q_i)_{i \in N}$, where $I_k = (G_k, N_k, \mathcal{U}_k)$ and $|N_k| = k$ (i.e., $I = I_n$). If $\mathcal{A}$ does not fail on any of these instances, then $\mathcal{A}(I, (q_i)_{i \in N})$ returns a desired allocation: each agent is allocated a bundle that she values at least as highly as her given value q_i. We need to prove that none of the recursive calls fails. To this end, we will prove the following lemma.

Algorithm 1: $\mathcal{A}(I', (q_i)_{i \in N'})$

input : $I' = (G', N', \mathcal{U}')$ and $(q_i)_{i \in N'}$ where G' is a subtree of G, N' is a subset of N, and $u'_i = u_i|_{V'}$ for all $i \in N'$

output : A valid allocation π such that $u_i(\pi(i)) \geqslant q_i$ for all $i \in N'$

1 **if** $u'_i(V') < q_i$ *for some $i \in N'$* **then**
2   **return** *fail*
3 **else if** $|N'| = 1$ **then**
4   **return** π *where $\pi(i) = V'$ for $\{i\} = N'$;*
5 **else**
6   Turn G' into a rooted tree;
7   Find $i \in N'$ and $v_i \in V'$ such that $u'_i(D(v_i)) \geq q_i$, but $u'_j(D(w)) < q_j$ for each child w of v_i and each player $j \in N'$;
8   Set $I'' = (G' \setminus D(v_i), N' \setminus \{i\}, \mathcal{U}'')$ where $\mathcal{U}''$ is given by $u''_j = u'_j|_{V' \setminus D(v_i)}$ for all $j \in N' \setminus \{i\}$;
9   **if** $\mathcal{A}(I'', (q_j)_{j \in N' \setminus \{i\}})$ *does not fail* **then**
10   Set $\pi' \leftarrow \mathcal{A}(I'', (q_j)_{j \in N' \setminus \{i\}})$;
11   Set $\pi(i) = D(v_i)$ and $\pi(j) = \pi'(j)$ for each $j \in N' \setminus \{i\}$;
12   **return** π;

Lemma 5.2. $\mathrm{mms}_j(I_k) \geqslant q_j$ *for all $k \in [n]$ and all $j \in N_k$.*

Proof. The proof proceeds by backwards induction on k. For $k = n$ the statement of the lemma is true. Suppose that the claim is true for some $k > 1$; we will prove it for $k-1$. Consider the player $i \in N_k \setminus N_{k-1}$, and let $D(v_i)$ be the bundle allocated to this player. For each player $j \in N_{k-1} = N_k \setminus \{i\}$ by the inductive hypothesis we have $\mathrm{mms}_j(I_k) \geqslant q_j$. Consider a partition $\mathcal{P} = (P_1, \ldots, P_k)$ witnessing this; $u_j(P_\ell) \geqslant q_j$ for each $\ell \in [k]$. Assume without loss of generality that $v_i \in P_1$. Then $D(v_i)$ is fully contained in P_1: if there is a vertex w in $D(v_i) \setminus P_1$, then the part of $\mathcal{P}$ that contains w is fully contained in a subtree rooted at a child of v_i, and hence the value of that part is strictly less than q_j, a contradiction.

Now, if $P_1 \setminus D(v_i)$ is not empty, then it is a subtree of G, and there is another part $P \in \mathcal{P}$ that is adjacent to $P_1 \setminus D(v_i)$ in G. Therefore, $\mathcal{P}' = (\mathcal{P} \setminus \{P_1\}) \cup \{P \cup (P_1 \setminus D(v_i))\}$ is a partition of G_{k-1} into $k-1$ connected components. By construction, $u_j(P') \geqslant q_j$ for each $P' \in \mathcal{P}'$, which proves that $\mathrm{mms}_j(I_{k-1}) \geqslant q_j$. □

Now, consider what happens when $\mathcal{A}$ is called on I_k and $(q_i)_{i \in N_k}$ for some $k \in [n]$. Let $G_k = (V_k, E_k)$. We have $u_i(V_k) \geqslant \mathrm{mms}_i(I_k) \geqslant q_i$ for all $i \in N_k$, which implies that the algorithm does not fail. This completes the proof. □

Proposition 5.1 relies on being given $(q_i)_{i \in N}$ as its input, so we still need to show that MMS values on trees can be computed efficiently. It turns out that this can be accomplished by the same recursive algorithm. We note that for the general problem (without graph constraints, or equivalently, on complete graphs), computing MMS values is NP-hard, though they can be well-approximated [Woeginger, 1997].
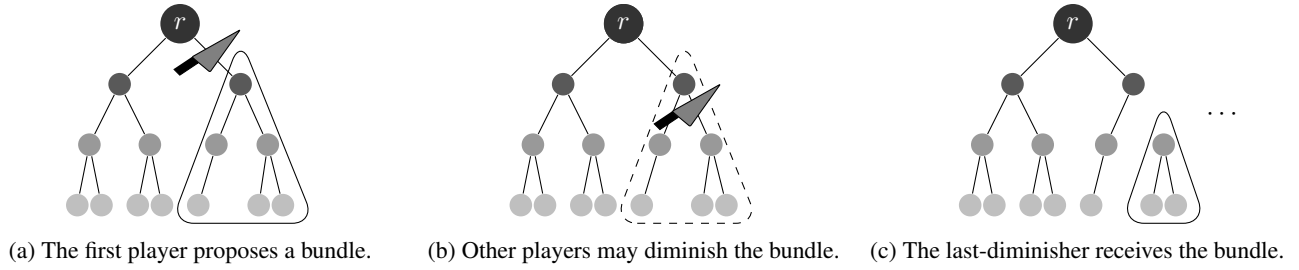
(a) The first player proposes a bundle.   (b) Other players may diminish the bundle.   (c) The last-diminisher receives the bundle.

Figure 2: A discrete version of the last-diminisher method

Lemma 5.3. *For an instance of CFD where G is a tree, and a player $i \in N$, we can compute $\mathrm{mms}_i(I)$ in polynomial time.*

Proof sketch. Calculating $\mathrm{mms}_i(I)$ is the same as maximizing the worst payoff for the instance I'' where all players are copies of player i. That is, $\mathrm{mms}_i(I)$ is equal to the maximum value of q such that $\mathcal{A}$ does not fail on $(I'', (q, \ldots, q))$. By scaling up the utilities, we can ensure that this value of q is a not-too-large integer (we omit the arithmetical details), and can therefore be found by binary search. $\qquad \square$

It now follows from Proposition 5.1 and Lemma 5.3 that for trees an MMS allocation can be computed efficiently.

Theorem 5.4. *Every instance $I = (G, N, \mathcal{U})$ of CFD where G is a tree admits an MMS allocation. Moreover, such an allocation can be computed in polynomial time.*

The known examples of instances without MMS allocations are very intricate. Our graph-based setting allows for simpler constructions: our next example shows that an MMS allocation may not exist on a cycle of 8 vertices. We conjecture that this is the shortest cycle that admits such an example. Our example is similar in spirit to an example for 2-additive utility functions by [Bouveret and Lemaître, 2015].

Example 5.5. Consider an instance $I = (G, N, \mathcal{U})$ of CFD where $G = (V, E)$ with $V = \{v_i \mid i = 1, 2, \ldots, 8\}$, $E = \{\{v_i, v_{i+1}\} \mid i = 1, 2, \ldots, 7\} \cup \{\{v_1, v_8\}\}$, $N = \{1, 2, 3, 4\}$, and the utilities are given as follows.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
Players 1 & 2	1	4	4	1	3	2	2	3
Players 3 & 4	4	4	1	3	2	2	3	1

To normalize to 1, each utility value above is divided by 20. Now, we have $\mathrm{mms}_1(I) = \mathrm{mms}_2(I) \geqslant 1/4$, as witnessed by the partition $P_1 = \{\{v_1, v_2\}, \{v_3, v_4\}, \{v_5, v_6\}, \{v_7, v_8\}\}$, which offers value $1/4$ for these players. Similarly, we have $\mathrm{mms}_3(I) = \mathrm{mms}_4(I) \geqslant 1/4$, as witnessed by the partition $P_2 = \{\{v_2, v_3\}, \{v_4, v_5\}, \{v_6, v_7\}, \{v_8, v_1\}\}$. These two partitions are illustrated below (note the cyclic shift):
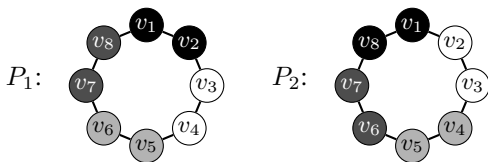
Now, suppose towards a contradiction that the instance I admits an MMS allocation π. Then π has to allocate at least two vertices to each player, as no player values any single item at $1/4$ or higher. This means that π partitions the cycle into either P_1 or P_2. Suppose first that π cuts the graph into P_1. Then, there is only one connected piece in P_1 that players 3 and 4 value at $1/4$ or higher, namely, $\{v_1, v_2\}$, so at least one of these players is allocated a piece whose value is less than his maximin share. A similar argument holds when π cuts the graph into P_2. Therefore, there is no MMS allocation. $\qquad \square$

6 Conclusions and Future Work

There are several exciting directions for the study of connected fair division of indivisible goods. For the solution concepts we have studied in this paper, one can ask whether certain graph classes yield better approximations than the general case, both in terms of existence guarantees and complexity results. In particular, it would be interesting to obtain a characterization of graphs for which an MMS allocation is guaranteed to exist. There are also further solution concepts that we have not considered, most notably the maximum Nash welfare solution [see Caragiannis *et al.*, 2016], which could be studied in this context both from the axiomatic and the computational points of view. Another promising direction would be to extend the work to other preference representations, including ordinal preferences [Aziz *et al.*, 2015], or to chores instead of goods [e.g., Aziz *et al.*, 2017]. Also, it would be interesting to obtain analogues of procedures such as sequential allocation and round-robin that respect the connectivity constraints and still produce desirable allocations. Finally, we may consider placing constraints on the 'shapes' of players' pieces, e.g., by requiring that the size of each piece is large relative to its diameter; similar ideas have been recently explored by [Segal-Halevi *et al.*, 2015] for the land division problem.

References

[Abebe *et al.*, 2017] R. Abebe, J. M. Kleinberg, and D. C. Parkes. Fair division via social comparison. In *AAMAS'17*, pages 281–289, 2017.

[Aumann and Dombb, 2015] Y. Aumann and Y. Dombb. The efficiency of fair division with connected pieces. *ACM Transactions on Economics and Computation*, 3:23:1–23:16, 2015.

[Aumann *et al.*, 2013] Y. Aumann, Y. Dombb, and A. Hassidim. Computing socially-efficient cake divisions. In *AAMAS'13*, pages 343–350, 2013.

[Aumann *et al.*, 2015] Y. Aumann, Y. Dombb, and A. Hassidim. Auctioning time: Truthful auctions of heterogeneous divisible goods. *ACM Transactions on Economics and Computation*, 4(1):3:1–3:16, 2015.

[Aziz and De Keijzer, 2011] H. Aziz and B. De Keijzer. Complexity of coalition structure generation. In *AAMAS'11*, pages 191–198, 2011.

[Aziz *et al.*, 2015] H. Aziz, S. Gaspers, S. Mackenzie, and T. Walsh. Fair assignment of indivisible objects under ordinal preferences. *Artificial Intelligence*, 227:71–92, 2015.

[Aziz *et al.*, 2017] H. Aziz, G. Rauchecker, G. Schryen, and T. Walsh. Algorithms for max-min share fair allocation of indivisible chores. In *AAAI'17*, pages 335–341, 2017.

[Bei *et al.*, 2012] X. Bei, N. Chen, X. Hua, B. Tao, and E. Yang. Optimal proportional cake cutting with connected pieces. In *AAAI'12*, pages 1263–1269, 2012.

[Bouveret and Lemaître, 2015] S. Bouveret and M. Lemaître. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Autonomous Agents and Multi-Agent Systems*, 30(2):259–290, 2015.

[Bouveret *et al.*, 2016] S. Bouveret, Y. Chevaleyre, and N. Maudet. Fair allocation of indivisible goods. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 12. Cambridge University Press, 2016.

[Brams and Taylor, 1996] S. J. Brams and Alan D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.

[Brânzei *et al.*, 2016] S. Brânzei, Y. Lv, and R. Mehta. To give or not to give: Fair division for single minded valuations. In *IJCAI'16*, pages 123–129, 2016.

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

[Caragiannis *et al.*, 2016] I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang. The unreasonable fairness of maximum nash welfare. In *ACM EC'16*, pages 305–322, 2016.

[Cechlárová and Pillárová, 2012] K. Cechlárová and E. Pillárová. On the computability of equitable divisions. *Discrete Optimization*, 9:249–257, 2012.

[Cechlárová *et al.*, 2013] K. Cechlárová, J. Doboš, and E. Pillárová. On the existence of equitable divisions. *Information Sciences*, 228:239–245, 2013.

[Chevaleyre *et al.*, 2017] Y. Chevaleyre, U. Endriss, and N. Maudet. Distributed fair allocation of indivisible goods. *Artificial Intelligence*, 242:1–22, 2017.

[Conitzer *et al.*, 2004] V. Conitzer, J. Derryberry, and T. Sandholm. Combinatorial auctions with structured item graphs. In *AAAI'04*, pages 212–218, 2004.

[Demko and Hill, 1998] S. Demko and T. P. Hill. Equitable distribution of indivisible items. *Mathematical Social Sciences*, 16:145–158, 1998.

[Garey and Johnson, 1979] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[Kurokawa *et al.*, 2016] D. Kurokawa, A. D. Procaccia, and J. Wang. When can the maximin share guarantee be guaranteed? In *AAAI'16*, pages 523–529, 2016.

[Lipton *et al.*, 2004] R. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *ACM EC'04*, pages 125–131, 2004.

[Procaccia and Wang, 2014] A. D. Procaccia and J. Wang. Fair enough: Guaranteeing approximate maximin shares. In *ACM EC'14*, pages 675–692, 2014.

[Segal-Halevi *et al.*, 2015] E. Segal-Halevi, A. Hassidim, and Y. Aumann. Envy-free cake-cutting in two dimensions. In *AAAI'15*, pages 1021–1028, 2015.

[Shrot *et al.*, 2010] T. Shrot, Y. Aumann, and S. Kraus. On agent types in coalition formation problems. In *AAMAS'10*, pages 757–764, 2010.

[Stromquist, 1980] W. Stromquist. How to cut a cake fairly. *American Mathematical Monthly*, 87:640–644, 1980.

[Stromquist, 2008] W. Stromquist. Envy-free divisions cannot be found by finite protocols. *The Electronic Journal of Combinatorics*, 15:145–158, 2008.

[Woeginger, 1997] G. J. Woeginger. A polynomial-time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters*, 20(4):149–154, 1997.