Fair Division of a Graph into Compact Bundles

Jayakrishnan Madathil
University of Glasgow, UK
jayakrishnan.madathil@glasgow.ac.uk

Abstract

We study the computational complexity of fair division of indivisible items in an enriched model: there is an underlying graph on the set of items. And we have to allocate the items (i.e., the vertices of the graph) to a set of agents in such a way that (a) the allocation is fair (for appropriate notions of fairness) and (b) each agent receives a bundle of items (i.e., a subset of vertices) that induces a subgraph with a specific "nice structure." This model has previously been studied in the literature with the nice structure being a connected subgraph. In this paper, we propose an alternative for connectivity in fair division. We introduce compact graphs, and look for fair allocations in which each agent receives a compact bundle of items. Through compactness, we attempt to capture the idea that every agent must receive a bundle of "closely related" items. We prove a host of hardness and tractability results with respect to fairness concepts such as proportionality, envy-freeness and maximin share guarantee.

1 Introduction

We study a fair allocation problem: m indivisible items are to be allocated among n agents, where the items are the vertices of a graph. The agents have preferences over the items, and we have to allocate the items in such a way that the allocation is fair (for appropriate notions of fairness) and each agent receives a bundle of items (i.e., a subset of vertices) that induces a subgraph with a specific "nice structure." The fairness notions that we study are proportionality, envy-freeness and maximin share guarantee; and the nice structure that we study requires that each agent must receive a set of "closely related" items.

While fair division of indivisible items has been studied extensively from both economics and computational perspectives [Moulin, 2003; Aziz *et al.*, 2022; Amanatidis *et al.*, 2022], fair division of graphs is a relatively recent line of study, starting with the work of [Bouveret *et al.*, 2017]. The classic fair division literature assumes no relationship among the items; the items are independent of each other. But there are any number of scenarios where the items to be allocated, rather than being an assortment of isolated units, are related to one another. And we can often model such relationships between the items using a graph. The items, for example, may be rooms in a large building, and the graph models adjacency between pairs of rooms; or the items may be a valuable collection of postage stamps, and the graph models similarities between pairs of stamps on the basis of country, design or typography; or the items may be the topics for a course (to be divided between two professors who are planning to co-teach the course), and the graph models connections between the topics. Fair division of graphs attempts to capture precisely such settings.

While fair division of graphs has received considerable attention in recent years [Deligkas *et al.*, 2021; Bilò *et al.*, 2022; Greco and Scarcello, 2020; Igarashi and Peters, 2019; Bei *et al.*, 2022], to the best of our knowledge, all work so far has focused solely on *connected* fair division. That is, each agent should receive a set of vertices that induces a connected subgraph. Bouveret et al. [2017], who introduced this model of fair division of graphs, considered the example of office spaces in a university being allocated to various research groups, where each research group should receive a contiguous set of offices. This scenario can be modelled as a fair-division-of-a graph problem where each agent must get a connected subgraph—agents represent various research groups and the vertices and the edges of the graph respectively represent the offices and adjacency between offices. Notice that while contiguity might be useful, the "closeness" of the offices allocated to each group might be just as important. Simply demanding contiguity might leave a research group with a set of offices along a long and narrow corridor, which may not be an attractive proposition to the members of the group. In the absence of other constraints, the connectivity requirement, while desirable, might produce bundles with unwieldy topologies. In some cases, demanding connected bundles for every agent may be too stringent a requirement. For example, imagine a scenario with just one agent and two items that are isolated from each other. The agent has a utility of $1/2$ for each of the two items and a total utility of 1 for the two items together. As we can see, demanding connectivity results in only one item being allocated to the agent, costing her half her total utility for the set of items.

We can also think of the connectivity requirement in the fair division of graphs as a direct adaptation of the contiguity requirement in the fair division of divisible items, i.e., the cake cutting problem where each agent should receive a contiguous piece of the cake [Stromquist, 1980; Goldberg *et al.*, 2020]. But contiguity is a singularly important consideration in the cake cutting setting, for otherwise an agent "who hopes only for a modest interval of cake may be presented instead with a countable union of crumbs" [Stromquist, 1980]. That is not the case with graphs. Moreover, graphs are a rich combinatorial structure capable of modelling a variety of relationships among the items. And connectivity of the items allocated to each agent may not always be the most important consideration. There is, however, a void when it comes to our understanding of fair division of graphs under constraints other than connectivity. In this paper, we take a first step towards filling this void.

1.1 Our Contribution

We make three contributions. (1) Propose meaningful alternatives for connectivity in the fair division of graphs. (2) Study the computational complexity of finding fair allocations under such alternative constraints, even on restricted inputs. The restrictions on the input are based on the nature of the valuation functions and the structure of the graph on the set of items. (3) Identify and exploit results and techniques from structural and algorithmic graph theory to design efficient algorithms for fair division of graphs. To that end, we propose an alternative for connectivity: We demand that each agent must receive a *compact* bundle of items. Through compactness, we try to capture the idea that each agent must receive a set of closely related items.

To formally define compact graphs, we introduce the following notation. For $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, 2, \ldots, n\}$ and $[n]_0 = [n] \cup \{0\}$. For a graph G, $V(G)$ and $E(G)$ respectively denote the set of vertices and edges of G. For vertices $z, z' \in V(G)$, the distance between z and z', denoted by $\text{dist}_G(z, z')$, is the length of a shortest path between z and z'. For $z \in V(G)$ and a non-negative integer β, $B_G(z, \beta) = \{z' \in V(G) \mid \text{dist}_G(z, z') \leq \beta\}$; we call $B_G(z, \beta)$ the ball of radius β centred at z.

Compact graphs: Definitions. For non-negative integers α and β, we say that a graph G is (α, β)-*compact* if there exist vertices $z_1, z_2, \ldots, z_\alpha \in V(G)$ such that for every $z \in V(G)$, $\text{dist}(z, z_i) \leq \beta$ for some $i \in [\alpha]$, i.e., $V(G) = \bigcup_{i=1}^{\alpha} B_G(z_i, \beta)$.[1] That is, G is (α, β)-compact if G can be "covered by α many balls, each of radius at most β." We also define a related class of graphs called strongly (α, β)-compact graphs. A graph G is *strongly* (α, β)-*compact* if there exist vertex subsets $V_1, V_2, \ldots, V_\alpha \subseteq V(G)$ such that $V(G) = \bigcup_{i=1}^{\alpha} V_i$ and for every $i \in [\alpha]$ and for every $z, z' \in V_i$, $\text{dist}_G(z, z') \leq \beta$. In this paper, we focus primarily on (α, β)-compact graphs. (We consider the empty graph—the graph G with $V(G) = E(G) = \emptyset$—to be both (α, β)-compact and strongly (α, β)-compact for every $\alpha, \beta \geq 0$.)

Compact graphs: Examples. If a graph G is strongly (α, β)-compact, then it also (α, β)-compact. But the converse need not hold. For example, a star is $(1, 1)$-compact, but not strongly $(1, 1)$-compact. But if G is (α, β) compact, then it is strongly $(\alpha, 2\beta)$-compact. When $\beta = 0$, the two definitions coincide. For example, a graph is G is (strongly) $(\alpha, 0)$-compact if and only if G has at most α many vertices. And G is strongly $(1, 1)$-compact if and only if G is a clique; and G is strongly $(1, \beta)$-compact if and only if G has diameter at most β. In particular, an m-vertex graph G is connected if and only if G is (strongly) $(1, m-1)$-compact. Similarly, a graph G with α connected components is strongly (α, β)-compact if and only if every component has diameter at most β. As these examples show, (strong) compactness is expansive enough to accommodate several natural restrictions on the structure of G, including properties of graphs such as number of vertices, connectivity, bounded diameter etc.

[1] In graph theory literature, a set $Z \subseteq V(G)$ of vertices such that $V(G) = \bigcup_{z \in Z} B_G(z, \beta)$ is called a distance-β dominating set [Haynes *et al.*, 2020]. So (α, β)-compact graphs are precisely those graphs that have a distance-β dominating set of size at most α.

Our Model

We study fair division problems of the following type. A typical instance of our problem consists of a triplet $(G, N, \mathcal{V})$, where G is an m-vertex graph, N is a set of agents with $|N| = n$, and $\mathcal{V}$ is an n-tuple of functions $(v_i)_{i \in N}$, where $v_i : 2^{V(G)} \to \mathbb{Q}_{\geq 0}$. For $i \in N$, v_i is called the valuation function or the utility function of agent i. We assume throughout that the valuation functions are additive (unless otherwise stated). That is, for $i \in N$ and $S \subseteq V(G)$, we have $v_i(S) = \sum_{z \in S} v_i(\{z\})$ and $v_i(\emptyset) = 0$. When $S = \{z\}$ is a singleton set, we omit the braces and simply write $v_i(z)$. We call the set of vertices of G goods or items as well. Throughout the paper, we use m for the number of items and n for the number of agents. And we use N or $[n]$ for the set of agents.

Compact allocations. Consider $(G, N, \mathcal{V})$. An allocation is a function that assigns pairwise disjoint subsets of $V(G)$ to the agents. That is, an allocation is a function $\pi : N \to 2^{V(G)}$ such that $\pi(i) \cap \pi(j) = \emptyset$ for distinct $i, j \in N$. Consider an allocation π. We call $\pi(i)$ agent i's bundle under the allocation π. We say that $\pi(i)$ is (α, β)-compact if $G[\pi(i)]$ is (α, β)-compact. An allocation π is (α, β)-*compact* if $\pi(i)$ is (α, β)-compact for every $i \in N$. We define a strongly (α, β)-compact allocation analogously.

Fairness concepts. We consider three well-studied fairness concepts—proportionality, envy-freeness and maximin fairness. An allocation π is *proportional* if $v_i(\pi(i)) \geq (1/n) \cdot v_i(V(G))$ for every $i \in N$, i.e., every agent receives at least a $1/n$ fraction of her utility for the whole graph. And we say that π is *envy-free* if $v_i(\pi(i)) \geq v_i(\pi(j))$ for every pair of distinct agents $i, j \in N$, i.e., every agent prefers her own bundle to that of the other agents. We now define maximin fair allocations by suitably adapting the definition introduced by Budish [2011]. For $(G, N, \mathcal{V})$, let $\Pi(G, N, \mathcal{V})$ be the set of all allocations $\phi : N \to 2^{V(G)}$. Let $\Gamma \subseteq \Pi(G, N, \mathcal{V})$ be any non-empty set of allocations. For every agent $i \in N$, we define the Γ-maximin share guarantee of i, denoted by $\Gamma\text{-}\texttt{mms}_i(G, N, \mathcal{V})$, as follows:

$$\Gamma\text{-}\texttt{mms}_i(G, N, \mathcal{V}) = \max_{\pi \in \Gamma} \min_{j \in N} v_i(\pi(j)).$$

Informally, $\Gamma\text{-}\texttt{mms}_i(G, N, \mathcal{V})$ is the maximum utility agent i could guarantee for herself if agent i were to allocate the items, with the caveat that the allocation be in Γ, and allowed to choose only the least valued bundle for herself. We say that an allocation $\phi : N \to 2^{V(G)}$ is Γ-*maximin fair* if $v_i(\pi(i)) \geq \Gamma\text{-}\mathtt{mms}_i(G, N, \mathcal{V})$. We are interested in Γ-maximin fair allocations, where Γ is the set of all (α, β)-compact allocations, which we denote by $\Gamma((\alpha, \beta)\text{-}\mathtt{com})$.

Economic efficiency constraints. We say that an allocation π is complete if π allocates all the vertices of G, i.e., $\bigcup_{i \in N} \pi(i) = V(G)$. And we say that π is Pareto-optimal if there exists no allocation π' such that $v_i(\pi'(i)) > v_i(\pi(i))$ for some $i \in N$ and $v_j(\pi'(j)) \geq v_j(\pi(j))$ for every $j \in N \setminus \{i\}$.

Computational questions. Let α and β be fixed integers. We are interested in computational problems that take an instance $(G, N, \mathcal{V})$ as input, and the question is to decide if $(G, N, \mathcal{V})$ admits an allocation that is (strongly) (α, β)-compact and fair? Depending on the fairness concept, we have the following specific problems.

- PROP-(α, β)-COMPACT-FD: Does $(G, N, \mathcal{V})$ admit an allocation that is proportional and (α, β)-compact?
- EF-(α, β)-COMPACT-FD: Does $(G, N, \mathcal{V})$ admit an allocation that is envy-free and (α, β)-compact? As the empty allocation (where no item is allocated) is trivially envy-free, we often combine envy-freeness with efficiency constraints such as completeness or Pareto-optimality (PO). Accordingly, we have the corresponding problems: COMPLETE-EF-(α, β)-COMPACT-FD and PO-EF-(α, β)-COMPACT-FD.
- MMS-(α, β)-COMPACT-FD: Does $(G, N, \mathcal{V})$ admit an allocation that is $\Gamma((\alpha, \beta)\text{-}\mathtt{com})$-maximin fair and (α, β)-compact?

We define the strongly compact variants of the problems analogously. Notice that α and β are fixed constants and not part of the input.

Our Results

We prove a host of hardness and algorithmic results for [X]-(α, β)-COMPACT-FD, where we use [X] as a placeholder for one of the three fairness concepts discussed above. Recall that n is the number of agents and m is the number of vertices in the input graph. We first discuss the polynomial time solvability versus (weak and strong) NP-hardness of the problems. These results are further divided into cases based on the choices for α and β. The hardness results hold only for problems corresponding to proportionality and envy-freeness, and they all hold for additive valuations. And then we move on to our algorithmic results, specifically, FPT and XP algorithms w.r.t. a combination of parameters, including the number of agents, maximum degree and treewidth of the graph. We now list our results one by one.

NP-hardness. A reduction due to Lipton et al. [2004] from the PARTITION problem shows that PROP-(α, β)-COMPACT-FD, COMPLETE-EF-(α, β)-COMPACT-FD and PO-EF-(α, β)-COMPACT-FD are all NP-hard for every $\alpha, \beta \geq 1$. But notice that a reduction from PARTITION only implies weak NP-hardness. Besides, these results do not cover the case when $\beta = 0$.

The case of $\beta = 0$. This case warrants special attention. In this case, every agent can receive at most α items as a graph is $(\alpha, 0)$-compact if and only if it has at most α vertices. (So we can ignore the graph and treat the problem as an instance of the classic fair division problem with the additional constraint on the size of the bundles.) We have the following results.

1. If $\alpha = 1$, then every agent can receive at most one item. In this case, for all three fairness notions, the corresponding problems are polynomial time solvable. We can reduce each of the problems to a matching problem in an agents-items bipartite graph. For envy-freeness, we have the the additional constraint that every agent must receive exactly one item; and the polynomial time algorithm is due to Gan et al. [2019].
2. The case of $\alpha = 2$. PROP-$(2, 0)$-COMPACT-FD is NP-hard; implied by a result due to Ferraioli et al. [2014]. The complexity of the problem corresponding to envy-freeness is still open.
3. The case of $\alpha \geq 3$ is covered by the next result.

The case of $\alpha \geq 3$. We prove that PROP-(α, β)-COMPACT-FD, COMPLETE-EF-(α, β)-COMPACT-FD and PO-EF-(α, β)-COMPACT-FD are all strongly NP-hard for every $\alpha \geq 3$ and $\beta \geq 0$, even when G is edgeless.

Proportionality + ($\alpha = 1$) + paths. If $\alpha = 1$ and the graph G is a path, then for proportionality, the corresponding problem admits an algorithm with runtime $2^n \cdot m^{\mathcal{O}(1)}$, and hence is fixed-parameter tractable (FPT) with respect to the number of agents. We also extend this result to an algorithm with runtime $n^p \cdot m^{\mathcal{O}(1)}$, where p is the number of *types* of agents. Two agents are of the same type if their valuations are identical. These results hold for arbitrary valuations. Our algorithms involve dynamic programming, and are only a slight adaptation of the algorithm of Bouveret et al. [2017] for proportional and *connected* fair division. But the fact that $\alpha = 1$ is crucial for these algorithms to work.

XP algorithms w.r.t. $n + \Delta$. For all three fairness concepts (and arbitrary valuations), we design slicewise polynomial time (XP) algorithms with respect to the parameter $n + \Delta$, where Δ is the maximum degree of G. The algorithms follow from the simple observation that for any vertex $z \in V(G)$, $|B_G(z, \beta)| \leq \Delta^{\beta+1}$. Hence we can enumerate all (α, β)-compact allocations in time $m^{\mathcal{O}(\alpha n)} \cdot 2^{\alpha n \cdot \Delta^{\mathcal{O}(\beta)}}$.

Pseudo-XP algorithms w.r.t. n + tw. For problems corresponding to all three fairness fairness concepts, we design pseudo-XP algorithms with respect to the parameter $\mathsf{tw} + n$, where tw is the treewidth of G. We design a single dynamic programming procedure that works for all three fairness concepts. We assume here that the valuations are integer-valued. Let W be the maximum value any agent has for the whole graph, i.e., $W = \max_{i \in N} v_i(V(G))$. Our algorithms have a runtime of $(\mathsf{tw} + n)^{\mathcal{O}(\mathsf{tw}+n)} \cdot \beta^{\mathcal{O}(\mathsf{tw})} \cdot m^{\mathcal{O}(\alpha n)} \cdot W^{\mathcal{O}(n^2)}$. In particular, when the valuations are polynomially bounded, we have XP algorithms. These algorithms yield a number of interesting corollaries, which we discuss below.

1. **Welfare maximisation:** We can use our dynamic programming procedure to compute (α, β)-compact alloca-

tions that maximise welfare. For example, we can answer questions such as this: Among all the allocations that maximise the utilitarian social welfare, is there one that is also (α, β)-compact? The utilitarian social welfare of an allocation π is the sum of the utilities of the agents have under π, i.e., $\sum_{i \in N} v_i(\pi(i))$.

2. **Connected fair division:** As noted earlier, an m-vertex graphs is connected if and only if it is $(1, m-1)$ compact. Hence our algorithms work for connected fair division as well, with a runtime of $(\mathsf{tw}+n)^{\mathcal{O}(\mathsf{tw}+n)} m^{\mathcal{O}(\mathsf{tw}+n)} \cdot W^{\mathcal{O}(n^2)}$.
3. **Implications for planar graphs (and more):** On planar graphs, our algorithms run in time $n^{\mathcal{O}(n)} \cdot m^{\mathcal{O}(\alpha n)} \cdot W^{\mathcal{O}(n^2)}$, i.e., a pseudo-XP algorithm parameterized by n. To derive this result, we leverage the follwing two facts. (1) Treewidth of a planar graph is $\mathcal{O}(D)$, where D is the diameter of the graph. (2) Diameter of every connected component of an (α, β)-compact graph is $\mathcal{O}(\alpha\beta)$. In fact, a runtime of $n^{\mathcal{O}(n)} \cdot m^{\mathcal{O}(\alpha n)} \cdot W^{\mathcal{O}(n^2)}$ is possible not just on planar graphs, but on all graphs whose treewidth is bounded by a function of its diameter. In particular, if $\mathcal{F}$ is a minor-closed family of graphs and $\mathcal{F}$ excludes an apex-graph, then on any graph $G \in \mathcal{F}$, our algorithms run in time $f(n) \cdot m^{\mathcal{O}(\alpha n)} \cdot W^{\mathcal{O}(n^2)}$, where the function f depends only on $\mathcal{F}$. This follows from a result due to Eppstein [2000].

Results for strongly compact variants. All our hardness results mentioned above hold for the strongly compact variants of the respective problems as well. So do (i) our algorithm for proportional allocations on paths when $\alpha = 1$ and (ii) the XP algorithms parametrized by $n + \Delta$. In addition, we show that for proportionality and envy-freeness + completeness, the corresponding strongly compact variants of the problems are strongly NP-hard for every $\alpha, \beta \geq 1$. (Notice that our strong NP-hardness results for the compact variants only cover the case of $\alpha \geq 3$.)

Runtime of our algorithms. The runtimes of the algorithmic results discussed above have an exponential dependence on n (the number of agents). But this need not be a disqualifying factor as several fair division settings only have a constant number of agents, and often, just two agents. Common examples of 2-agent settings, as noted by Plaut and Roughgarden [2020], include divorce settlements, inheritance division and international border disputes. In fact, 2-agent setting is one of the most intensively studied special cases in fair division; see, for example, [Brams and Fishburn, 2000; Brams *et al.*, 2022]. We would also like to point out that the dependence of the runtime of our treewidth-based algorithm on both n and tw is also essential. Our NP-hardness results for proportional or envy-free, compact allocations when $\alpha \geq 3$ hold for graphs of treewidth zero (even when the valuations are additive and polynomially bounded). The runtime of our treewidth-based algorithm depends also on W, the maximum valuation of an agent for the whole graph. This may be inevitable as well. As noted above, we can use our algorithms to find welfare-maximising allocations. Our algorithms thus compute allocations with three properties: fairness, compactness and economic efficiency. And for algorithms that find efficient allocations, "pseudo" runtimes are rather common [Barman *et al.*, 2018; Aziz *et al.*, 2023].

Choice of parameters. We would also like to emphasise that the parameterizations that we use to design our algorithm are all commonly studied in the literature. In particular, parameterizations by the number of agents or agent types have been used by Bouveret et al. [2017] and Deligkas et al. [2021]. It is quite common in parameterized algorithms literature to leverage structural parameters such as maximum degree, degeneracy or treewidth to design efficient algorithms. Treewidth, in particular, is a popular structural parameter; see, for example, [Fomin *et al.*, 2020] for the many applications of treewidth in algorithm design. We must also add that connected fair division of graphs has previously been studied on special graphs such as paths, stars, trees, cycles etc., which are all graphs of treewidth 1 or 2. Also, parameters such as treewidth and cliquewidth have been used as structural parameters in connected fair division. We discuss many of these works below. So it is very much in line with the existing literature to use treewidth as a parameter to design algorithms. In particular, one of our goals was to apply results and techniques from graph theory literature to problems in fair division. And for this purpose, treewidth is arguably the most suitable choice.

1.2 Related Work

Modern fair division literature goes back at least to the pioneering work of Steinhaus [1948]. See the surveys by Lindner and Rothe [2016] for the divisible goods setting; and by Aziz et al. [2022] or by Amanatidis et al. [2022] for the indivisible goods setting.

There is also a large volume of literature on fair division under connectivity constraints. As for the cake cutting setting (i.e., divisible goods), Stromquist [1980] proved that an envy-free allocation of the cake into connected pieces exists, but there is no finite protocol that can find such an allocation [Stromquist, 2008]. Coming to the indivisible goods setting, the work of Bouveret et al. [2017], which formally introduced this line of study, contains several algorithmic and hardness results for connected fair division of a graph. They showed that finding a connected allocation that is envy-free or proportional is NP-hard, even when the graph is a path. On the other hand, a maximin allocation always exists and can be computed in polynomial time if the graph is a tree. They also designed several FPT and XP algorithms with respect to parameters such as number of agents and number of agent types, for special cases of the graph. This was followed by the work of Lonc and Truszczynski [2018], who studied the existence and complexity of maximin fair allocations on cycles. Among other results, they proved that when there are only three agents, there exists an allocation that guarantees each agent a constant fraction of her maximin share. Most related to our work is that of Deligkas et al. [2021], who undertook an in-depth study of the parameterized complexity of connected fair division. They established a host of hardness and algorithmic results—under various combinations of parameters such as the number of agents, treewidth, cliquewidth etc.—with respect to fairness concepts such as proportionality, envy-freeness, EF1 and EFX. A number of recent works deal with the special case when the graph is a path, under various fairness notions. In particular, Bilò et al. [2022] studied the existence and complexity of EF1 and EF2 allocations; Igarashi [2022] studied EF1 allocations and showed that EF1 allocations always exists for any number of agents with monotone valuations; Suksompong [2019] studied approximate guarantees for proportionality, envy-freeness and equitability; and Misra et al. [2021] studied the complexity of finding allocations that are equitable up to one item (EQ1) combined with economic efficiency notions such as Pareto-optimality, non-wastefulness etc. In addition to these works that mainly deal with the existence or computational questions associated with connected fair division, Bei et al. [2022] studied the price of connectivity—the loss incurred by agents when the connectivity constraint is imposed—for maximin fair allocations. Their techniques included the application of several graph theoretic tools and concepts to fair division.

Much of the literature on fair division under restricted input settings focus on restrictions on valuations, number of agents or relaxations of fairness notions. To the best of our knowledge, there have only been a handful of works that consider restrictions on the *structure* of bundles allocated to agents (other than connected fair division in the context of graphs). The typical restriction on bundles involves what can be called cardinality constraints. The canonical example is the house allocation problem, where each agent must receive exactly one item. Gan et al. [2019] studied envy-freeness and Kamiyama et al. [2021] studied proportionality and equitability in the house allocation setting. Ferraioli et al. [2014] et al. studied the problem of maximising egalitarian welfare, under the additional constraint that each agent be allocated exactly k items. Biswas and Barman [2018] introduced a more general variant of cardinality constraints: the items are partitioned into groups and there is a cap on each group's contribution to any agent's bundle. Among other results, they showed that a $1/3$-approximate maximin fair allocation can be computed efficiently. Recently, Hummel and Hetland [2022] improved the approximation guarantee to $1/2$. See the survey by Suksompong [2021] for a comprehensive discussion of various constraints in fair division.

Organisation of the paper. Due to space constraints, we present only a representative subset of results here. We sketch the proofs for an NP-hardness result and our treewidth DP.

2 Strong NP-Hardness of Compact FD

We show that checking if a proportional, (α, β)-compact allocation exists is strongly NP-hard when $\alpha \geq 3$.

Theorem 1. *For every $\alpha \geq 3$ and $\beta \geq 0$,* PROP-(α,β)-COMPACT-FD *is strongly NP-hard.*

Proof sketch. Fix $\alpha \geq 3$ and $\beta \geq 0$. We show the NP-hardness of PROP-(α,β)-COMPACT-FD by a reduction from EXACT COVER BY α-SETS (XαC), which is known to be NP-complete for every $\alpha \geq 3$ [Karp, 1972].[2] In XαC, the input consists of a set X such that $|X| = \alpha s$ and a family $\mathcal{F} \subseteq 2^X$ of subsets of X such that $|F| = \alpha$ for every $F \in \mathcal{F}$; and the question is to decide if $(X, \mathcal{F})$ has an exact cover, i.e., a sub-family $\mathcal{F}' \subseteq \mathcal{F}$ such that $|\mathcal{F}'| = s$ and $\bigcup_{F \in \mathcal{F}'} F = X$. Consider an instance $(X, \mathcal{F})$ of XαC. Let $|F| = r$ and $\mathcal{F} = \{F_1, F_2, \ldots, F_r\}$. We assume without loss of generality that $r > s$. Given $(X, \mathcal{F})$, we now construct an instance $(G, N, \mathcal{V})$ of PROP-(α,β)-COMPACT-FD as follows. Corresponding to each $F_i \in \mathcal{F}$, we introduce an agent i; in addition we introduce a "dummy" agent $r+1$. Thus $N = [r+1]$. Corresponding to each $x \in X$, we introduce an item w_x; in addition, we introduce $r - s$ "auxiliary" items $y_1, y_2, \ldots, y_{r-s}$ and a "special" item y_*. And we define the graph G to be the edge-less graph with vertex set $V(G) = \{y_*\} \cup \{y_j \mid j \in [r-s]\} \cup \{w_x \mid x \in X\}$. Finally, we define the valuation functions as follows. Consider $i \in [r]$. For each $x \in X$, we set

$$v_i(w_x) = \begin{cases} 1/(\alpha r + \alpha), & \text{if } x \in F_i \\ 0, & \text{if } x \notin F_i; \end{cases}$$

$v_i(y_j) = 1/(r+1)$ for every $j \in [r-s]$; and $v_i(y_*) = s/(r+1)$. Finally, $v_{r+1}(w_x) = 0$ for every $x \in X$; $v_{r+1}(y_j) = 0$ for every $j \in [r-s]$; and $v_{r+1}(y_*) = 1$. Notice that we have $\sum_{z \in V(G)} v_i(z) = 1$ for every $i \in N$. And notice also that since G is edge-less, for every non-empty subset $S \subseteq V(G)$, the subgraph $G[S]$ is (α, β)-compact if and only if $|S| \leq \alpha$. We can verify that $(X, \mathcal{F})$ is a yes-instance of XαC if and only if $(G, N, \mathcal{V})$ is a yes-instance of PROP-(α,β)-COMPACT-FD. □

3 Fair Division of Graphs of Bounded Treewidth

In this section, we show that for each of the three fairness concepts—proportionality, envy-freeness and maximin fairness—the corresponding problem of checking if there exists an allocation that is fair and (α, β) compact admits a pseudo-XP algorithm, when parameterized by the treewidth of the item graph and the number of agents. We design a single dynamic programming procedure that works for all three fairness concepts. **We assume that the valuations are integer-valued.** The following theorem is the main contribution of this section.

Theorem 2. *For every α and β,* PROP-(α,β)-COMPACT-FD, EF-(α,β)-COMPACT-FD *and* MMS-(α,β)-COMPACT-FD *admit algorithms that run in time* $(\mathsf{tw} + n)^{\mathcal{O}(\mathsf{tw}+n)} \cdot \beta^{\mathsf{tw}} \cdot m^{\mathcal{O}(\alpha n)} \cdot W^{\mathcal{O}(n^2)}$, *where* tw *is the treewidth of the input graph G and W is the maximum valuation of an agent for G, i.e., $W = \max_{i \in N} v_i(V(G))$.*

Annotated allocations. To prove theorem 2, we introduce an auxiliary problem. For a graph G, a vertex $\hat{z} \in V(G)$ and $\beta > 0$, we say that G is a $(\hat{z}; \beta)$-annotated graph if $z \in B_G(\hat{z}, \beta)$ for every $z \in V(G)$, i.e., every vertex in G is within a distance of at most β from $\hat{z}$. Consider a graph G, a set N of n agents and $\beta > 0$. For an n-tuple $(\hat{z}_i)_{i \in N}$ and an allocation $\pi : N \to 2^{V(G)}$, we say that π is $((\hat{z}_i)_{i \in N}; \beta)$-annotated if for every $i \in N$, $\hat{z}_i \in \pi(i)$ and

[2] Karp [1972] only shows the NP hardness of X3C. But it is straightforward to reduce X3C to XαC for every $\alpha > 3$.

$G[\pi(i)]$ is a $(\hat{z}_i;\beta)$-annotated graph. And we define the associated computational problem ANNOTATED-FD as follows: Given $(G,N,\mathcal{V})$, $(\hat{z}_i)_{i\in N}$ and β as input, does $(G,N,\mathcal{V})$ admit a $((\hat{z}_i)_{i\in N};\beta)$-annotated allocation?

Remark 3 (Reducing to ANNOTATED-FD)**.** We now show how to use annotated allocations to prove Theorem 2. Suppose $(G,N,\mathcal{V})$ is a yes-instance [X]-(α,β)-COMPACT-FD; and let $\pi : N \to 2^{V(G)}$ be the hypothetical fair and (α,β)-compact allocation that we are looking for. Then for every $i \in N$, $G[\pi(i)]$ is (α,β)-compact. Hence for every $i \in N$, there exist vertices $z_i^1, z_i^2, \ldots, z_i^\alpha$ such that $\pi(i) = \bigcup_{j=1}^{\alpha} B_{G[\pi(i)]}(z_i^j,\beta)$. **To design our algorithm, we first guess the vertices $z_i^1, z_i^2, \ldots, z_i^\alpha$ for every $i \in N$.** There are $m^{\mathcal{O}(\alpha n)}$ many guesses. For each guess, we do as follows. For every $i \in N$, we add a new vertex $\hat{z}_i$ and make it adjacent to $z_i^1, z_i^2, \ldots, z_i^\alpha$; and for every $j \in N$, we set $v_j(\hat{z}_i) = 0$. We then look for a fair and $((\hat{z}_i)_{i\in N}, \beta+1)$-annotated allocation. Notice that for any vertex $z \in V(G)$ and $i \in N, j \in [\alpha]$, z is at a distance of at most β from z_i^j if and only if z is at a distance of at most $\beta+1$ from $\hat{z}_i$. To summarise, given an instance of [X]-(α,β)-COMPACT-FD, we can construct $m^{\mathcal{O}(\alpha n)}$ instances of ANNOTATED-FD such that the original instance is a yes-instance if and only if at least one of the instances of ANNOTATED-FD is a yes-instance.

In light of Remark 3, to solve [X]-(α,β)-COMPACT-FD, it is enough to solve ANNOTATED-FD, for which we design an algorithm. More precisely, we prove the following lemma.

Lemma 4. *There is an algorithm that, given an instance $(G,[n],\mathcal{V},\beta,(\hat{z}_i)_{i\in[n]})$ of* ANNOTATED-FD*, an n^2 tuple $(\hat{w}_{ij})_{i,j\in[n]}$, where $0 \le \hat{w}_{ij} \le W$ for every $i,j \in [n]$ and a nice tree decomposition $(T,\{X_t \mid t \in V(T)\})$ of G as input, runs in time $(\mathsf{tw}+n)^{\mathcal{O}(\mathsf{tw}+n)} \cdot \beta^{\mathcal{O}(\mathsf{tw}+n)} \cdot m^{\mathcal{O}(1)} \cdot W^{\mathcal{O}(n^2)}$, and correctly decides if $(G,[n],\mathcal{V},\beta,(\hat{z}_i)_{i\in[n]})$ admits a $((\hat{z}_i)_{i\in[n]};\beta)$-annotated allocation $\pi : [n] \to 2^{V(G)}$ such that $v_i(\pi(j)) = \hat{w}_{ij}$ for each $i,j \in [n]$.*

The rest of this section is dedicated to sketching a proof of Lemma 4. And for that, we design a dynamic programming algorithm over a (nice) tree decomposition of G. We assume basic familiarity with tree decompositions and refer the reader to [Cygan *et al.*, 2015, Chapter 7] for an overview on tree decomposition based algorithms.

A tool for our DP: Rooted partition. Consider a non-empty set S. A partition of S is a family $\mathcal{P} \subseteq 2^S$ of non-empty subsets of S such that $\bigcup_{P\in\mathcal{P}} P = S$ and $P \cap P' = \emptyset$ for every distinct $P, P' \in \mathcal{P}$. For a partition $\mathcal{P}$ of S, each set $P \in \mathcal{P}$ is called a *block* of $\mathcal{P}$. And for $z \in S$, we denote the unique block of $\mathcal{P}$ that contains z by $\mathtt{block}_{\mathcal{P}}(z)$. We now define what we call a *rooted partition*. For a non-empty set S, a rooted partition $\mathcal{P}$ of S is nothing but a partition of S, but each block P of $\mathcal{P}$ has a designated element, which we call the root of P. Formally, a rooted partition of S is an ordered pair $(\mathcal{P}, R)$ such that $\mathcal{P}$ is a partition of S, $R \subseteq S$ with $|R| = |\mathcal{P}|$ and for each block P of $\mathcal{P}$, $|P \cap R| = 1$; we call the unique element of $P \cap R$ the root of P.

From now on, we assume that we are given $(G,[n],\mathcal{V},\beta,(\hat{z}_i)_{i\in[n]})$, an n^2 tuple $(\hat{w}_{ij})_{i,j\in[n]}$, where $0 \le \hat{w}_{ij} \le W$ for every $i,j \in [n]$ and a nice tree decomposition $(T,\{X_t \mid t \in V(T)\})$ of G of width tw. **We first add the vertices $\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_n$ to every bag X_t** (and denote the resulting tree decomposition by $(T,\{X_t \mid t \in V(T)\})$ as well). Notice that this increases the width of the tree decomposition by n; and $(T,\{X_t \mid t \in V(T)\})$ still remains a nice tree decomposition but for the fact that the bags corresponding to the root node and the leaf nodes of T are non-empty. In particular, for $t \in V(T)$, where t is a leaf node or t is the root of T, we have $X_t = \{\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_n\}$.

Outline of our algorithm. Suppose that π is the $((\hat{z}_i)_{i\in[n]};\beta)$-annotated allocation that we are looking for. Then $\mathrm{dist}_{G[\pi(i)]}(\hat{z}_i, z) \le \beta$ for each $i \in [n]$ and for each $z \in \pi(i)$. Let H_i be a BFS tree of $G[\pi(i)]$, rooted at $\hat{z}_i$. Since H_i is a BFS tree rooted at $\hat{z}_i$, we have $\mathrm{dist}_{G[\pi(i)]}(\hat{z}_i, z) = \mathrm{dist}_{H_i}(\hat{z}_i, z)$ for every $z \in \pi(i)$. For a node $t \in V(T)$, we guess how H_i intersects with the graph G_t. Notice that H_i is an acyclic graph and it could possibly split into multiple connected components when intersected with G_t. With this in mind, we guess the following features of the intersection of H_i with G_t: (a) the intersection of $V(H_i)(=\pi(i))$ with X_t, (b) how H_i partitions the vertices of $X_t \cap \pi(i)$ into connected components (c) for each $z \in X_t \cap \pi(i)$, the distance in H_i between z and $\hat{z}_i$ and (4) agent i's valuations for the subsets of goods in G_t that have been allocated to herself as well as the other agents, i.e., $v_i(\pi(j) \cap V(G_t))$ for every $i,j \in [n]$. We argue that these four pieces of information are sufficient to design a dynamic programming algorithm that constructs π in a bottom up fashion over T.

Designing the DP: Necessary Ingredients. To design our DP, for each node $t \in V(T)$, we first define a set $\mathcal{C}_t$, (which formalises our guesses). Formally, for each $t \in V(T)$, let $\mathcal{C}_t$ be the set of all tuples $((S_i, f_i, (\mathcal{P}_i, R_i))_{i\in[n]}, (w_{ij})_{i,j\in[n]})$ with the following properties.

- For each $i \in [n]$, S_i is a vertex subset such that $\hat{z}_i \in S_i$ and $S_i \subseteq X_t \cap B_G(\hat{z}_i,\beta)$ and $S_i \cap S_j = \emptyset$ for every $j \in [n] \setminus \{i\}$.
- For each $i \in [n]$, $(\mathcal{P}_i, R_i)$ is a rooted partition of S_i such that $\hat{z}_i \in R_i$.
- For each $i \in [n]$, $f_i : S_i \to [\beta]_0$ is a function such that $f_i(\hat{z}_i) = 0$ and $f_i(z) \ne 0$ for every $z \in S_i \setminus \{\hat{z}_i\}$.
- The tuple $(w_{ij})_{i,j\in[n]}$ is an n^2-tuple of non-negative integers such that $0 \le w_{ij} \le W$ for every $i,j \in [n]$.

Observation 5. *For each $t \in V(T)$, we have $|\mathcal{C}|_t \le (\mathsf{tw}+n)^{\mathcal{O}(\mathsf{tw}+n)} \cdot \beta^{\mathcal{O}(\mathsf{tw})} \cdot W^{\mathcal{O}(n^2)}$. To see this, observe the following facts that follow from the definition of $\mathcal{C}_t$. (i) As the sets $S_1, S_2, \ldots, S_n \subseteq X_t$ are pairwise disjoint, each element of X_t belongs to at most one S_i, and therefore, the number of choices for $(S_i)_{i\in[n]}$ is $(n+1)^{|X_t|}$. By identical reasoning, the number of choices for $(R_i)_{i\in[n]}$ is $(n+1)^{|X_t|}$. (ii) For each choice of $(S_i)_{i\in[n]}$, the family $\bigcup_{i=1}^n \mathcal{P}_i$ is a partition of $\bigcup_{i=1}^n S_i \subseteq X_t$. Therefore, corresponding to each $(S_i)_{i\in[n]}$, the number of choices for $(\mathcal{P}_i)_{i\in[n]}$ is at most $|X_t|^{|X_t|}$. (iii) As the sets $S_1, S_2, \ldots, S_n$ are pairwise disjoint, we may think of the n-tuple $(f_i)_{i\in[n]}$ of functions as a* *single function from $\bigcup_{i=1}^{n} S_i$ to $[\beta]_0$. Therefore, the number of choices for $(f_i)_{i\in[n]}$ is at most $(\beta+1)^{|X_t|}$. (iv) We have $|X_t| \le \mathsf{tw}+1+n$; the "$+n$" accounts for the fact that we added $\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_n$ to every bag. (v) The number of choices for the tuple $(w_{ij})_{i,j\in[n]}$ is at most $(W+1)^{n^2}$.*

Valid allocations. We now define what we call valid allocations. Informally, these are allocations that agree with our guesses (i.e., the tuples in $\mathcal{C}_t$). For a node $t \in V(T)$, let G_t be the subgraph of G made up of all the vertices and edges introduced in the subtree rooted at t. Consider a node $t \in V(T)$ and an allocation $\pi : [n] \to 2^{V(G_t)}$. For a tuple $\eta \in \mathcal{C}_t$, where $\eta = ((S_i, f_i, (\mathcal{P}_i, R_i))_{i\in[n]}, (w_{ij})_{i,j\in[n]})$, we say that π is (t,η)-*valid* if

(*VA*.1) for each $i \in [n]$, $\pi(i) \cap X_t = S_i$;

(*VA*.2) for each $i \in [n]$, $G_t[\pi(i)]$ has a spanning forest H_i such that
- (a) for every $z, z' \in S_i$, z and z' are in the same connected component of H_i if and only if z and z' are in the same block of $\mathcal{P}_i$;
- (b) for every $z \in S_i$, $f_i(z) = f_i(x) + \mathrm{dist}_{H_i}(x,z)$, where x is the root of $\mathtt{block}_{\mathcal{P}_i}(z)$;
- (c) for every $z \in \pi(i) \setminus X_t$, there exists $y \in R_i$ such that $\mathrm{dist}_{H_i}(y,z) \le \beta - f_i(y)$; and

(*VA*.3) for each $i, j \in [n]$, $v_i(\pi(j)) = w_{ij}$.

For a $t \in V(T), \eta \in \mathcal{C}_t$, a (t,η)-valid allocation $\pi : [n] \to 2^{V(G_t)}$ and $i \in [n]$, we call a spanning forest H_i of $G_t[\pi(i)]$ that satisfies conditions (*VA*.2)(a)–(*VA*.2)(c) a (t,η,i)-*witness for* π. The correctness of our DP crucially relies on the following lemma, which says that every connected component of a (t,η,i)-witness intersects X_t.

Lemma 6. *Consider $i \in [n]$. For $t \in V(T)$ and $\eta = ((S_i, f_i, (\mathcal{P}_i, R_i))_{i\in[n]}, (w_{ij})_{i,j\in[n]}) \in \mathcal{C}_t$, let $\pi : [n] \to 2^{V(G_t)}$ be a (t,η)-valid allocation and H_i a (t,η,i)-witness for π. Then for each connected component H of H_i, there exists a unique vertex $x_H \in V(H) \cap R_i$.*

Observation 7 ((Structure of a (t,η,i)-witness))**.** *For $i \in [n]$, we can think of a (t,η,i)-witness H_i as a rooted forest—each connected component is a rooted tree; R_i is precisely the set of roots of the components of H_i, each connected component of H_i intersects X_t and the partition $\mathcal{P}_i$ is such that for $z, z' \in S_i$, z and z' are in the same block of $\mathcal{P}_i$ if and only if z and z' are in the same connected component of H_i.*

Before proceeding further with our DP, let us first see how valid allocations help us prove Lemma 4.

Lemma 8. *Consider $(G, [n], \mathcal{V}, \beta, (\hat{z}_i)_{i\in[n]})$, $(\hat{w}_{ij})_{i,j\in[n]}$, and $(T, \{X_t \mid t \in V(T)\})$ of G as defined in Lemma 4. And consider an allocation $\pi : [n] \to 2^{V(G)}$. Then π is a $((\hat{z}_i)_{i\in[n]}; \beta)$-annotated allocation with $v_i(\pi(j)) = \hat{w}_{ij}$ for every $i, j \in [n]$ if and only if π is $(\hat{t}, \hat{\eta})$-valid, where $\hat{t}$ is the root of the tree T and $\hat{\eta} = ((S_i, f_i, (\mathcal{P}_i, R_i))_{i\in[n]}, (\hat{w}_{ij})_{i,j\in[n]}) \in \mathcal{C}_{\hat{t}}$ with $S_i = \{\hat{z}_i\}$, $f_i : S_i \to [\beta]_0$ is the function that maps $\hat{z}_i$ to 0, $P_i = \{\{\hat{z}_i\}\}$ and $R_i = \{\hat{z}_i\}$ for each $i \in [n]$.*

Proof. Assume first that π is a $((\hat{z}_i)_{i\in[n]}; \beta)$-annotated allocation and $v_i(\pi(j)) = \hat{w}_{ij}$ for every $i, j \in [n]$. We show that π satisfies each of the conditions in the definition of $(\hat{t}, \hat{\eta})$-valid allocation. Recall that $X_{\hat{t}} = \{\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_n\}$ and $G_{\hat{t}} = G$. Consider $i \in [n]$. First, $\pi(i) \cap X_{\hat{t}} = \pi(i) \cap \{\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_n\} = \{\hat{z}_i\} = S_i$. Thus π satisfies condition (*VA*.1). Since π is $((\hat{z}_i)_{i\in[n]}; \beta)$-annotated, $\mathrm{dist}_{G[\pi(i)]}(\hat{z}_i, z) \le \beta$ for every $z \in \pi(i)$. Now, fix a BFS tree H_i of $G[\pi(i)]$, rooted at $\hat{z}_i$. Since $\pi(i) \cap X_t = \{\hat{z}_i\}$ and $\mathcal{P}_i = \{\{\hat{z}_i\}\}$, H_i trivially satisfies condition (*VA*.2)(a). Again, since $S_i = \{\hat{z}_i\}$, $f_i(\hat{z}_i) = 0$ and $\mathrm{dist}_{H_i}(\hat{z}_i, \hat{z}_i) = 0$, H_i trivially satisfies condition (*VA*.2)(b) as well. Observe now that by the definition of a BFS tree, for every $z \in \pi(i)$, we have $\mathrm{dist}_{H_i}(\hat{z}_i, z) = \mathrm{dist}_{G[\pi(i)]}(\hat{z}_i, z) \le \beta$, which along with the fact that $f_i(\hat{z}_i) = 0$, implies that $\mathrm{dist}_{H_i}(\hat{z}_i, z) \le \beta - f_i(\hat{z}_i)$. Thus H_i satisfies condition (*VA*.2)(c). Finally, as $v_i(\pi(j)) = \hat{w}_{ij}$, π satisfies condition (*VA*.3). We have thus shown that π is $(\hat{t}, \hat{\eta})$-valid.

Conversely, assume that π is $(\hat{t}, \hat{\eta})$-valid. Consider $i \in [n]$. Recall that $S_i = \{\hat{z}_i\}$ and $f_i(\hat{z}_i) = 0$. By condition (*VA*.1), we have $\hat{z}_i \in \pi(i)$. By the definition of a $(\hat{t}, \hat{\eta})$-valid allocation, π has a (t,η,i)-witness, say H_i'. By condition (*VA*.2)(c), for every $z \in \pi(i) \setminus \{\hat{z}_i\}$, we have $\mathrm{dist}_{H_i'}(\hat{z}_i, z) \le \beta$, which implies that $\mathrm{dist}_{G[\pi(i)]}(\hat{z}_i, z) \le \beta$. Thus $((\hat{z}_i)_{i\in[n]}; \beta)$-annotated. □

Definition of the states of the DP. Lemma 8 tells us that to check if $(G, N, \mathcal{V})$ admits a $((\hat{z}_i)_{i\in[n]}; \beta)$-annotated allocation, it is sufficient to check if $(G, N, \mathcal{V})$ admits a $(\hat{t}, \hat{\eta})$-valid allocation. In light of this, we now define the states of our DP as follows. **For each $t \in V(T)$ and each tuple $\eta \in \mathcal{C}_t$, we define $A[t,\eta] = 1$ if there exists a (t,η)-valid allocation and $A[t,\eta] = 0$ otherwise.**

We omit the details of the computation of $A[t,\eta]$. We conclude with the observation that since the DP allows us to check if there exists an allocation π with $v_i(\pi(j)) = w_{ij}$ for all possible values of w_{ij} for every $i, j \in [n]$, it is straightforward to check if there exists an allocation that satisfies the required fairness constraints.

4 Conclusion

We proposed an alternative for connectivity in the fair division of graphs. Our results demonstrate that we can achieve tractability results under such alternative constraints. Our results also demonstrate that a number of tools and deep results from graph theory could find applications in fair division; this is still an under-explored direction. Compact allocations, we believe, can be a compelling alternative for connected allocations, and warrant further study. This work leaves several questions open. First, it would be interesting to see if our algorithmic results translate to the strongly compact setting. Second, all our strong NP-hardness results rely on an arbitrary number of agents and non-identical valuations. It would be interesting to see if strong NP-hardness results can be proved for restricted input settings. Third, apart from connectivity and compactness, there may be other structured bundles that are worth investigating. We hope this work will trigger such questions.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) [EP/V032305/1]. The author thanks the anonymous referees and Jessica Enright for their comments on the manuscript.

References

[Amanatidis *et al.*, 2022] Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, and Alexandros A. Voudouris. Fair division of indivisible goods: A survey. In *IJCAI*, pages 5385–5393. ijcai.org, 2022.

[Aziz *et al.*, 2022] Haris Aziz, Bo Li, Hervé Moulin, and Xiaowei Wu. Algorithmic fair allocation of indivisible items: A survey and new questions. *CoRR*, abs/2202.08713, 2022.

[Aziz *et al.*, 2023] Haris Aziz, Xin Huang, Nicholas Mattei, and Erel Segal-Halevi. Computing welfare-maximizing fair allocations of indivisible goods. *Eur. J. Oper. Res.*, 307(2):773–784, 2023.

[Barman *et al.*, 2018] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Finding fair and efficient allocations. In *EC*, pages 557–574. ACM, 2018.

[Bei *et al.*, 2022] Xiaohui Bei, Ayumi Igarashi, Xinhang Lu, and Warut Suksompong. The price of connectivity in fair division. *SIAM J. Discret. Math.*, 36(2):1156–1186, 2022.

[Bilò *et al.*, 2022] Vittorio Bilò, Ioannis Caragiannis, Michele Flammini, Ayumi Igarashi, Gianpiero Monaco, Dominik Peters, Cosimo Vinci, and William S. Zwicker. Almost envy-free allocations with connected bundles. *Games Econ. Behav.*, 131:197–221, 2022.

[Biswas and Barman, 2018] Arpita Biswas and Siddharth Barman. Fair division under cardinality constraints. In *IJCAI*, pages 91–97. ijcai.org, 2018.

[Bouveret *et al.*, 2017] Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, and Dominik Peters. Fair division of a graph. In *IJCAI*, pages 135–141. ijcai.org, 2017.

[Brams and Fishburn, 2000] Steven J. Brams and Peter C. Fishburn. Fair division of indivisible items between two people with identical preferences: Envy-freeness, pareto-optimality, and equity. *Soc. Choice Welf.*, 17(2):247–267, 2000.

[Brams *et al.*, 2022] Steven J. Brams, D. Marc Kilgour, and Christian Klamler. Two-person fair division of indivisible items when envy-freeness is impossible. *Oper. Res. Forum*, 3(2), 2022.

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

[Cygan *et al.*, 2015] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

[Deligkas *et al.*, 2021] Argyrios Deligkas, Eduard Eiben, Robert Ganian, Thekla Hamm, and Sebastian Ordyniak. The parameterized complexity of connected fair division. In *IJCAI*, pages 139–145. ijcai.org, 2021.

[Eppstein, 2000] David Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3):275–291, 2000.

[Ferraioli *et al.*, 2014] Diodato Ferraioli, Laurent Gourvès, and Jérôme Monnot. On regular and approximately fair allocations of indivisible goods. In *AAMAS*, pages 997–1004. IFAAMAS/ACM, 2014.

[Fomin *et al.*, 2020] Fedor V. Fomin, Stefan Kratsch, and Erik Jan van Leeuwen, editors. *Treewidth, Kernels, and Algorithms - Essays Dedicated to Hans L. Bodlaender on the Occasion of His 60th Birthday*, volume 12160 of *Lecture Notes in Computer Science*. Springer, 2020.

[Gan *et al.*, 2019] Jiarui Gan, Warut Suksompong, and Alexandros A. Voudouris. Envy-freeness in house allocation problems. *Math. Soc. Sci.*, 101:104–106, 2019.

[Goldberg *et al.*, 2020] Paul Goldberg, Alexandros Hollender, and Warut Suksompong. Contiguous cake cutting: Hardness results and approximation algorithms. *J. Artif. Intell. Res.*, 69:109–141, 2020.

[Greco and Scarcello, 2020] Gianluigi Greco and Francesco Scarcello. The complexity of computing maximin share allocations on graphs. In *AAAI*, pages 2006–2013. AAAI Press, 2020.

[Haynes *et al.*, 2020] Teresa W Haynes, Stephen T Hedetniemi, and Michael A Henning. *Topics in domination in graphs*, volume 64. Springer, 2020.

[Hummel and Hetland, 2022] Halvard Hummel and Magnus Lie Hetland. Maximin shares under cardinality constraints. In *EUMAS*, volume 13442 of *Lecture Notes in Computer Science*, pages 188–206. Springer, 2022.

[Igarashi and Peters, 2019] Ayumi Igarashi and Dominik Peters. Pareto-optimal allocation of indivisible goods with connectivity constraints. In *AAAI*, pages 2045–2052. AAAI Press, 2019.

[Igarashi, 2022] Ayumi Igarashi. How to cut a discrete cake fairly. *CoRR*, abs/2209.01348, 2022.

[Kamiyama *et al.*, 2021] Naoyuki Kamiyama, Pasin Manurangsi, and Warut Suksompong. On the complexity of fair house allocation. *Oper. Res. Lett.*, 49(4):572–577, 2021.

[Karp, 1972] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[Lindner and Rothe, 2016] Claudia Lindner and Jörg Rothe. Cake-cutting: Fair division of divisible goods. In *Economics and Computation*, Springer texts in business and economics, pages 395–491. Springer, 2016.

[Lipton *et al.*, 2004] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *EC*, pages 125–131. ACM, 2004.

[Lonc and Truszczynski, 2018] Zbigniew Lonc and Miroslaw Truszczynski. Maximin share allocations on cycles. In *IJCAI*, pages 410–416. ijcai.org, 2018.

[Misra *et al.*, 2021] Neeldhara Misra, Chinmay Sonar, P. R. Vaidyanathan, and Rohit Vaish. Equitable division of a path. *CoRR*, abs/2101.09794, 2021.

[Moulin, 2003] Hervé Moulin. *Fair division and collective welfare*. MIT Press, 2003.

[Plaut and Roughgarden, 2020] Benjamin Plaut and Tim Roughgarden. Communication complexity of discrete fair division. *SIAM J. Comput.*, 49(1):206–243, 2020.

[Steinhaus, 1948] Hugo Steinhaus. The problem of fair division. *Econometrica*, 16:101–104, 1948.

[Stromquist, 1980] Walter Stromquist. How to cut a cake fairly. *The American Mathematical Monthly*, 87(8):640–644, 1980.

[Stromquist, 2008] Walter Stromquist. Envy-free cake divisions cannot be found by finite protocols. *Electron. J. Comb.*, 15(1), 2008.

[Suksompong, 2019] Warut Suksompong. Fairly allocating contiguous blocks of indivisible items. *Discret. Appl. Math.*, 260:227–236, 2019.

[Suksompong, 2021] Warut Suksompong. Constraints in fair division. *SIGecom Exch.*, 19(2):46–61, 2021.