Fair and Efficient Division of a Discrete Cake with Switching Utility Loss

AAAI Track

Zheng Chen
College of Computer Science and Technology
Zhejiang University
Hangzhou, China
21721122@zju.edu.cn

Bo Li
Department of Computing
The Hong Kong Polytechnic University
Hong Kong, China
comp-bo.li@polyu.edu.hk

Minming Li
Department of Computer Science
City University of Hong Kong
Hong Kong, China
minming.li@cityu.edu.hk

Guochuan Zhang
College of Computer Science and Technology
Zhejiang University
Hangzhou, China
zgc@zju.edu.cn

ABSTRACT

Cake cutting is a widely studied model for allocating resources with temporal or spatial structures among agents. Recently, a new line of research has emerged that focuses on the discrete variant, where the resources are indivisible and connected by a path. In some real-world applications, the resources are interdependent, and dividing the cake may reduce their effectiveness. In this paper, we introduce a model that captures the effect of division as switching utility loss and investigate the tradeoff between fairness and efficiency for various settings. Specifically, we measure fairness and efficiency using the popular notions of envy-freeness up to one item (EF1) and social welfare, respectively. The goal of our study is to understand how much social welfare must be sacrificed to ensure EF1 allocations and design polynomial-time algorithms that can compute EF1 allocations with the best possible social welfare guarantee.

KEYWORDS

Fair division; Envy-freeness; Social welfare

ACM Reference Format:
Zheng Chen, Bo Li, Minming Li, and Guochuan Zhang. 2024. Fair and Efficient Division of a Discrete Cake with Switching Utility Loss: AAAI Track. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

1 INTRODUCTION

Since the very beginning of fair resource allocation research, cake cutting has been studied as a canonical model, where a cake, denoted by the real interval $[0, 1]$, is to be allocated to n agents who have heterogeneous preferences over different parts of the cake [2, 11, 25]. Various solution concepts have been proposed to measure the fairness of an allocation, and envy-freeness (EF) is among the most widely accepted ones [15]. Informally, an allocation is EF if every agent gets the best piece of the cake compared with other agents evaluated by her own preference. An EF allocation always exists [11] and can be found in finite steps [2, 3]. One of the various real-life applications of the cake-cutting problem is to fairly schedule time slots for, for example, online meetings. People, possibly at different times zones, may have different perspectives on their preferred time slots, where time is abstracted as the cake $[0, 1]$. One key feature of the cake is that it can be arbitrarily divided at any position. However, as noted by some recent works [5, 17, 18, 21, 26], this assumption may not hold in practice, including the previous time scheduling problem. For example, time is usually partitioned into discrete time slots by hours or half hours, and thus it can only be divided at the boundaries of these slots. This problem is called *discrete cake cutting*, where the cake is modelled as a path, i.e., a line of vertices (representing the items) that are connected by edges. In the following, for discrete cake cutting problems, we turn to use *path* to refer to the cake.

Motivated by different real-life applications, several variants of the discrete cake cutting problem have been studied. For example, Bouveret et al. [9] and follow up works such as [8, 17] imposed connectivity constraints on the allocations, which requires every agent to receive a set of contiguous vertices. The continuity constraints is in part motivated by the fact that the division of the cake may suffer utility loss, and imposing continuity is one simple way to increase the effectiveness of the allocations. Actually, in many situations, the utility loss of the division can be quantified. For instance, when a family doctor (or tutor, housekeeper, etc.) is scheduled to help with patients A and B between 1 pm and 2 pm and between 3 pm and 4 pm as illustrated in Figure 1, nobody gets the utility from time slot between 2 pm and 3 pm since the doctor needs to spend time travelling from A's place to B's. When considering the usage of a hall, the time slot in the middle of two events, which is for cleaning or setup, will be lost. Motivated by these examples, in this paper, we consider the discrete cake cutting problem when the utilities at the switching points decrease to 0.

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

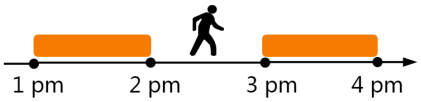

Figure 1: The example of moving from one event to another.

Enforcing the allocations to be fair may sacrifice the efficiency, such as social welfare, inevitably. To understand the effect of fairness, Bertsimas et al. [7] and Caragiannis et al. [13] proposed the notion of *price of fairness* (PoF), which quantifies the worst case ratio between the optimal social welfare without and with fairness constraints, and later Bei et al. [6] and Barman et al. [4] bounded the PoF for indivisible items under various fairness notions. The objectives of our paper are two-fold, namely analyzing the PoF for the model with switching utility loss and designing polynomial-time algorithm to compute fair allocations with the guaranteed social welfare approximation. Higher PoF indicates that the fairness concept is incompatible with efficiency. In this case, we shall identify valuable cases where we manage to ensure both fairness and efficiency simultaneously.

1.1 Our Model and Main Results

We model the cake as a path, where the vertices are the items to be allocated, and the agents have possibly different weight functions on the *edges*.[1] Upon receiving a set of vertices, an agent's utility is decided by the total weight of the edges in the induced subgraph. Note that one key feature of this model is that the agents' utilities are determined by the combination of the items, which can be viewed as a new class of valuation functions in the standard indivisible fair division problems. Similar to the literature, since an EF allocation may not exist, our fairness notion is the *envy-freeness up to one item* (EF1), which requires that an agent does not envy another agent if one vertex is removed from the latter's bundle. The existence (and efficient computation) of an EF1 allocation is proved by Lipton et al. [20]. However, an arbitrary EF1 allocation may not be efficient in terms of social welfare. Our objective is to design algorithms for various settings to compute EF1 allocations with high social welfare guarantee.

Our main results are summarized in Table 1. We first observe that no algorithm can guarantee the social welfare of EF1 allocations to be better than $1/n$ fraction of the optimal social welfare without any constraints. Then we move to some important and commonly studied cases for which the efficiency can be significantly improved. When there are two agents or when the weight functions are binary, we design polynomial-time algorithms to compute EF1 allocations that can always achieve at least half of the optimal social welfare. Further, if both conditions hold (i.e., two agents with binary weight functions), the social welfare guarantee can be improved to $2/3$. We also consider the homogeneous case when the agents have identical weight functions. Actually, the approximation of $2/3$ cannot be improved, but can be (almost) achieved no matter how many agents and what weight functions we have. Note that all these approximations are proven to be (almost) the best possible and thus imply the (almost) tight bounds for PoF regarding EF1 fairness.

In comparison, for the classic additive setting when the values are on the items instead of the edges, Bei et al. [6] and Barman et al. [4] proved that the tight bound of PoF regarding EF1 is $\Theta(1/\sqrt{n})$ for scaled valuations and is $\Theta(1/n)$ for unscaled valuations. When $n = 2$, it is between 0.865 and 0.875. Obviously, for the homogeneous or binary case, the PoF is 1.

Finally, motivated by the graphic structures of the items, we generalize our model to graphs beyond paths. Our approaches can be extended to trees and similar results hold. However, when the graphs are arbitrary, we find significant difference. First, in trees, computing a social welfare maximizing allocation can be done in polynomial time, but it is NP-hard in general graphs. Second, for general graphs, even when the agents have identical and binary weight functions, ensuring better than $\Omega(1/n)$ fraction of the optimal social welfare by EF1 allocations is not possible.

1.2 Related Works

Cake-Cutting. There are a rich line of works investigating the cake-cutting problem particularly on ensuring distinct fairness notions. The classic cake-cutting problem dates back to the seminal work of [24] in the 1940s. The existence of an envy-free allocation is guaranteed [1], even with $n - 1$ cuts [14]. Brams and Taylor [11] presented how to constructively find them, although subtle complexity questions remain open [23]. With intense efforts over decades, Aziz and Mackenzie [3] proposed the discrete and bounded EF protocol, which was later extended to the general case for n agents [2].

Discrete Cake-Cutting. The prominent model in the discrete cake-cutting literature assumes that the cake is represented by a path, which has been considered in [12, 21, 26]. Assume that no two agents prefer the same items placed at the same position, Marenco and Tetzlaff [21] proved the existence of an envy-free allocation for any discrete cake and any number of players. Similar problems on sequences were analyzed in [12]. Considering the non-existence of allocation with the classic fairness notions such as proportionality, envy-freeness and equitability, Suksompong [26] investigated the approximate counterpart of these notions as well as quantifying the loss of efficiency.

Graph-Based Fair Division. Also closely related are the works of fair allocation with graph constraints [9, 10, 18, 19]. Bouveret et al. [9] initiated the research concerning allocations of indivisible items under graph-based constraints, and showed that it is NP-hard to find a complete envy-free allocation. Igarashi and Peters [18] focused on allocations guaranteeing Pareto-optimality under connectivity constraints. They showed the hardness of deciding whether a Pareto-optimal EF1 connected allocation exists, even when the graph is a path or the weights are binary and additive. Bouveret et al. [10] dealt with the allocation of chores and showed the hardness results concerning achieving envy-free or equitable solutions. Igarashi and Zwicker [19] proposed the model of tangled cake where multiple copies of the unit intervals are glued together. They explored the effect of tangles on achieving EF allocations of

[1] Equivalently, we can regard the edges as the items to be allocated, but we then need to impose extra constraints such that if an edge connects two edges that are allocated to different agents, it must be disposed and cannot be allocated to anyone.

	heterogeneous agents				homogeneous agents
	arbitrary weights		binary weights		n agents & arbitrary weights
	n agents	2 agents	n agents	2 agents	
approx. ratio	$\Theta(1/n)$ (Thm 3.1)	$1/2$ (Thm 5.1)	$1/3$ (Thm 6.1)	$2/3$ (Thm 5.2)	$\approx 2/3$ (Thm 4.1 and Thm 4.4)

Table 1: Main Results, where "$\approx 2/3$" means "$2/3$" when $n = 2$ and "$2/3 - o(1)$" when $n \geq 3$.

connected shares and showed that exactly six tangles guarantee EF fairness. Besides envy-freeness, other classic fairness notions-*maximin share* (aka MMS) and *proportionality* (aka PROP) have been studied extensively [5, 9, 16, 27]

2 PRELIMINARIES

Denote an instance by $\mathcal{I} = (G, N, \mathbf{w})$ (also rewrite $\mathcal{I} = (G, N)$ for short for homogeneous cases), where $G = (V, E)$ is a path, modeling the cake to be distributed among a set of agents $N = \{1, \ldots, n\}$. $V = \{v_1, \ldots, v_m\}$ is a set of m vertices (i.e., the items to be allocated), ordered from left to right, that are connected by edges in E, i.e., v_j and v_{j+1} are connected by an edge $e_j = (v_j, v_{j+1}) \in E$ for all $1 \leq j < m$. Throughout this paper, n and m are reserved for the numbers of agents and items. Each agent i has a weight function $w_i : E \to \mathbb{R}_{\geq 0}$ on the edges. Let $\mathbf{w} = \{w_1, \ldots, w_n\}$. The function w_i is called binary if $w_i(e) \in \{0, 1\}$ for all $e \in E$. For any bundle of items $X \subseteq V$, let $G[X]$ be the corresponding induced subgraph of X within the original graph G. Given any subgraph G', denote by $V(G')$ and $E(G')$ the sets of vertices and edges of G'. The agents' weight functions induce their utility functions. In particular, for each agent $i \in N$, her utility function is the mapping $u_i : 2^V \to \mathbb{R}^+ \cup \{0\}$ such that $u_i(X)$ for $X \subseteq V$ equals the weight of all edges within $G[X]$, i.e., $u_i(X) = \sum_{e \in E(G[X])} w_i(e)$. When the agents have identical weight functions, we omit the subscript and directly write $w(\cdot)$ and $u(\cdot)$.

Let $\Pi_n(V)$ be the set of all n-partitions of V. An allocation $\mathbf{X} = (X_1, \ldots, X_n)$ corresponds to a n-partition in $\Pi_n(V)$, where $X_i \subseteq V$ contains the items allocated to agent $i \in N$. An allocation $\mathbf{X}$ is called partial if $\bigcup_{i \in N} X_i \subsetneq V$. Our main solution concept is the following EF1 fairness.

Definition 2.1 (EF1). *An allocation $\mathbf{X} = (X_1, \ldots, X_n)$ is called* envy-free up to 1 item *(EF1) if for any i and j with $X_j \neq \emptyset$, there exists $g \in X_j$ such that $u_i(X_i) \geq u_i(X_j \setminus \{g\})$.*

The *social welfare* of allocation $\mathbf{X} = (X_1, \ldots, X_n)$ is $\mathrm{sw}(\mathbf{X}) = \sum_{i \in N} u_i(X_i)$, and the optimal social welfare of an instance $\mathcal{I}$ is denoted by

$$\mathrm{sw}^*(\mathcal{I}) = \max_{\mathbf{X} \in \Pi_n(V)} \sum_{i \in N} u_i(X_i).$$

If the instance $\mathcal{I}$ is clear from the context, we also write sw^* for short.

Lipton et al. [20] proposed the *envy-cycle elimination* (ECE) algorithm, which always returns an EF1 allocation, as long as the agents' utility functions are monotone. Informally, given a (partial) allocation $(X_1, \ldots, X_n)$, we can construct the corresponding *envy graph* $\mathcal{G} = (N, \mathcal{E})$, where the nodes are agents (and thus agents and nodes are used interchangeably) and there is a directed edge from agent i to agent j if and only if $u_i(X_i) < u_i(X_j)$. The ECE algorithm runs as follows. We first find an agent who is not envied by the others, and allocate a new item to her. If there is no such agent, there must be a cycle in $\mathcal{G}$. Then we resolve this cycle by reallocating the bundles: every agent gets the bundle of the agent that she envies in the cycle. We keep resolving cycles until there is an unenvied agent. Repeat the above procedures until all the items are allocated. Note that if the agents have identical valuations, then the ECE algorithm degenerates to a simple greedy algorithm where the worst-off agent selects a new item without creating any envy cycles. For simplicity, we continue to call this greedy algorithm ECE.

Although ECE algorithm ensures EF1, the returned allocation does not have any social welfare guarantee. Consider the following simple example. A path of three vertices $v_1 - v_2 - v_3$ is distributed among two agents with identical weight functions. By ECE, we allocate v_1 to agent 1 , then v_2 to agent 2 and finally v_3 to agent 1. The social welfare is $u(\{v_1, v_3\}) = u(v_2) = 0$. However, the optimal social welfare is 2 by allocating $\{v_1, v_2, v_3\}$ to agent 1, which is also an EF1 allocation. Thus, our objective is to design algorithms to compute EF1 allocations with guaranteed social welfare approximation ratio compared to the optimal social welfare without constraints.

Thanks to the ECE algorithm, to compute an EF1 allocation with high social welfare, we can first find a partial allocation that is EF1 and ensures the guaranteed social welfare. Then we can run the ECE algorithm to allocate the remaining unallocated items, where the social welfare can only increase and the EF1 property retains.

Before the end of this section, we show that the maximum social welfare can be computed in polynomial time via dynamic programming. Suppose the vertices in $V = \{v_1, \ldots, v_m\}$ are ordered from left to right, and thus v_i and e_i are the i-th vertex and i-th edge. For subproblem $1 \leq k \leq m$, let $\mathrm{sw}(k)$ and $\mathbf{X}^k = (X_1^k, \ldots, X_n^k)$ respectively denote the maximum social welfare and the corresponding partial allocation when allocating vertices $\{v_1, \ldots, v_k\}$ among the n agents. Denote $\mathrm{sw}(0) = 0$ and it is also clear that $\mathrm{sw}(1) = 0$. Our objective is to iteratively compute each $\mathrm{sw}(k)$ for $k \geq 2$ until $\mathrm{sw}(m)$. For $1 \leq l \leq k-1$, let a'_l be the agent who receives the l-th item in the partial welfare maximizing allocation $\mathbf{X}^l$, then we have the following Bellman equation:

$$\mathrm{sw}(k) = \max_{1 \leq l \leq k-1} \max \left\{ \mathrm{sw}(l) + \sum_{t=l}^{k-1} w_{a'_l}(e_t), \mathrm{sw}(l-1) + \max_{i \in N} \sum_{t=l}^{k-1} w_i(e_t) \right\}.$$

In the appendix, we provide a formal proof why $\mathrm{sw}(m)$ returns the optimal social welfare in polynomial time. We summarize the result in the following lemma.

Lemma 2.2. *Given any instance $\mathcal{I} = (G, N, \mathbf{w})$, computing an allocation with maximum social welfare can be done in polynomial time.*

In the appendix, we also show that the dynamic programming algorithm for Lemma 2.2 can be extended to tree structures. However, for general graphs, computing such an allocation is NP-hard.

3 GENERAL SETTING

We start with the general case, but due to space limit, the results are formally proven in the appendix. In the following, we briefly introduce our ideas. We first find that EF1 allocations cannot ensure better than $1/n$ fraction of the optimal social welfare, even for the bi-valued case. This is not surprising; consider a simple example with n independent edges where one of the agents, say agent 1, has weight 1 for all edges but all the other agents have arbitrarily small weights. To ensure EF1, at most one edge can be allocated to agent 1, resulting in a social welfare of approximately 1 but the optimal social welfare is n. This hard instance is because of the fact that the agents' valuations are not in the same scale. Actually, for EF1 allocations under scaled valuations, the approximation ratio of the optimal social welfare can be improved to $\Theta(1/\sqrt{n})$.

Note that when the graph is a set of disjoint edges, the problem degenerates (with slight differences) to the classical additive setting, for which it is NP-hard to find the EF1 allocations with optimal social welfare. Therefore, we now briefly discuss how to achieve a good approximation of the optimal social welfare by EF1 allocations. Since the agents' valuations depend on their weights on the edges and the vertices can only be allocated once, the immediate intuition is to pair up the vertices as independent edges and treat the edges as items. But we need to be careful. Consider the example in Figure 2 with two agents and agent 1's weight function is shown above the edges while agent 2's is below the edges. Suppose the vertices are paired as (v_1, v_2), (v_3, v_4) and (v_5, v_6), and some algorithm allocates edges (v_1, v_2) and (v_5, v_6) to agent 1. At this moment, agent 2 does not envy agent 1, then the algorithm may further allocate (v_3, v_4) to agent 1; however, this breaks the EF1 requirement since v_3 and v_4 connect v_2 and v_5 to form edge (v_2, v_3) and (v_4, v_5), on which agent 2 has high weights.

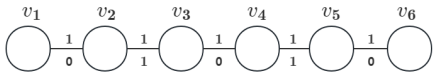

Figure 2: Hard instance for allocating edges.

To address the above issue, we can give up one vertex between two consecutive pairs of vertices. For example, the vertices can be paired as (v_1, v_2), (v_4, v_5), (v_7, v_8), etc., and vertices v_3, v_6, etc., are abandoned. In this way, we can regard these pairs as imaginary items whose addition does not affect others (the bad situation in Figure 2 will not happen) and the removal of one endpoint in each pair destroys the edge. Thus running a standard EF1 algorithm (for additive valuations), such as ECE, gives an EF1 (partial) allocation in our problem. To extend the partial allocation to a complete allocation, we can continue to run the ECE algorithm on the abandoned vertices.

Since nobody gains utility at the broken edge between two vertices, to ensure high social welfare of the allocation, we would like to allocate relatively longer sub paths (instead of independent edges) in order to reduce the number of divisions. Since we need to maintain the property that the removal of one vertex also removes the entire sub path, we can group every three consecutive vertices, where each triplet is an imaginary item, such as (v_1, v_2, v_3). Then the removal of the middle vertex in each triplet destroys both incident edges, and thus removes the complete sub path.

Combining the above two ideas, we consider two partitions of the vertices, $M_1 = \{(v_1, v_2, v_3), (v_5, v_6, v_7), \ldots\}$ and $M_2 = \{(v_3, v_4, v_5), (v_7, v_8, v_9), \ldots\}$. It can be shown that the partition with higher social welfare can ensure at least a half of the optimal social welfare. Then we can use this partition to allocate its triplets and our problem essentially degenerates to the classic additive setting. In particular, we run the ALG-EF1-ABS algorithm proposed in [4] which ensures $1/(2n)$ fraction of the optimal social welfare, and thus we have the following result.

Theorem 3.1. *Given any discrete cake cutting instance $\mathcal{I}$, an EF1 allocation with social welfare at least $1/(4n) \cdot \mathrm{sw}^*(\mathcal{I})$ can be computed in polynomial time; There is an instance $\mathcal{I}$ such that no EF1 allocation can guarantee a social welfare better than $1/n \cdot \mathrm{sw}^*(\mathcal{I})$.*

Similarly, we can obtain the result for the scaled valuations, proved in the appendix.

Theorem 3.2. *Given any discrete cake cutting instance $\mathcal{I}$ with scaled valuations, an EF1 allocation with social welfare at least $\Theta(1/\sqrt{n}) \cdot \mathrm{sw}^*(\mathcal{I})$ can be computed in polynomial time.*

In the following sections, we show how to improve the approximation ratios for some specific, yet important, settings.

4 IMPROVED APPROXIMATION FOR HOMOGENEOUS AGENTS

In this section, we consider the homogeneous case when all agents have identical valuations (but the weights can be arbitrary). As we will see, we can always compute an EF1 allocation guaranteeing almost 2/3 fraction of the optimal social welfare in polynomial time, and 2/3 is the best guarantee we can obtain even if the valuations are binary and there are only two agents.

The algorithm is formally introduced in Algorithm 1. Similarly, it tries to partition the items into bundles and regards each bundle as an imaginary item. However, different from the algorithm designed in Theorem 3.1, we do not want to give up any vertex between two bundles since the utilities on the incident edges would be very likely sacrificed. Thus we partition the vertices into contiguous triplets such as as (v_1, v_2, v_3), (v_4, v_5, v_6), (v_7, v_8, v_9), etc. Recall the bad situation discussed in Figure 2. If we happen to allocate two adjacent triplets to the same agent who is currently the worst-off one, the resulting allocation may not be EF1, and thus the main difficulty in Algorithm 1 is how to avoid such bad situations.

Before proceeding, we first briefly discuss our ideas. Suppose i is the worst-off agent in the current round and the remaining triplets are $\{\Theta_j, \Theta_{j+1}, \ldots\}$ ordered from left to right. We would like

Algorithm 1 Approximation Algorithm for n Homogeneous Agents

Input: Instance $\mathcal{I} = (G, N)$ with $G = (V, E; w)$.
Output: Allocation $\mathbf{X} = (X_1, \ldots, X_n)$.

Initialize $X_i \leftarrow \emptyset$ for all $i \in N$.
Consider the three sets of triplets M_1, M_2, M_3 defined in Equation (1), and let $M \leftarrow \arg\max_{M \in \{M_1, M_2, M_3\}} u(V(M_i))$ be the one with maxium social welfare.
Denote by $M = \{\Theta_1, \ldots, \Theta_{|M|}\}$ and assume the triplets in M are ordered from left to right.
Let i^* be the last agent who receives a triplet in M in previous rounds. Initialize $i^* \leftarrow 0$.
Let P contain the triplets that are adjacent to the vertices allocated to the agent who has smallest utility. Initialize $P \leftarrow \emptyset$.
Let Q contain the vertices in a triplet that is going to be broken. Initialize $Q \leftarrow \emptyset$.
// Phase 1: Allocating the triplets in M.
while $M \neq \emptyset$ **do**
  Let i be the agent with smallest utility.
  Case 1(a). If $i \neq i^*$ and $P \neq \emptyset$, let Θ be one triplet in P. Set $X_i \leftarrow X_i \cup \{\Theta\}$ and $P \leftarrow P \setminus \{\Theta\}$.
  Case 1(b). If $i \neq i^*$ and $P = \emptyset$, let Θ be the first triplet in M. Set $X_i \leftarrow X_i \cup \{\Theta\}$, $M \leftarrow M \setminus \{\Theta\}$, and $i^* = i$.
  Case 1(c). If $i = i^*$, move the first triplet in M to P. If M still contains at least one triplet, let Θ be the first one. Set $X_i \leftarrow X_i \cup \{\Theta\}$ and $M \leftarrow M \setminus \{\Theta\}$.
end while
// If $P \neq \emptyset$, then all triplets in P can only be adjacent to the vertices allocated to agent i^*.
// Phase 2: Allocating the triplets in P.
while $P \neq \emptyset$ **do**
  Let i be the agent with smallest utility.
  Case 2(a). If $i \neq i^*$, let Θ be one triplet in P. Set $X_i \leftarrow X_i \cup \{\Theta\}$ and $P \leftarrow P \setminus \{\Theta\}$.
  Case 2(b). If $i = i^*$ and $Q \neq \emptyset$, let v be one vertex in Q. Set $X_i \leftarrow X_i \cup \{v\}$ and $Q \leftarrow Q \setminus \{v\}$.
  Case 2(c). If $i = i^*$ and $Q = \emptyset$, remove one triplet Θ^* in P and put all its vertices to Q. Let v be one vertex in Q, and set $X_i \leftarrow X_i \cup \{v\}$ and $Q \leftarrow Q \setminus \{v\}$.
end while
// If $Q \neq \emptyset$, then the vertices in Q are from the same triplet Θ^*.
// Phase 3: Allocating the vertices in Q.
if $Q \neq \emptyset$ and $u(V(\Theta^*)) \geq u(X_{i^*} \setminus V(\Theta^*))$ **then**
  Set $Q \leftarrow X_{i^*} \setminus V(\Theta^*)$ and $X_{i^*} \leftarrow V(\Theta^*)$.
end if
Execute ECE on Q.
return Allocation $\mathbf{X} = (X_1, \ldots, X_n)$.

to allocate the first triplet Θ_j to i, but i already has Θ_{j-1} in her bundle, which is adjacent to Θ_j in the path. Then, we create a pool P to temporarily store Θ_j, and give Θ_{j+1} to agent i. In later rounds, we always try to allocate the triplets in P first and if $P = \emptyset$, we go back to check the path. When all the triplets have been either allocated to the agents or stored in P, we complete the first phase, which is the first **while** loop in Algorithm 1. Note that if $P \neq \emptyset$, then we can show that all triplets in P can only be adjacent to the vertices allocated to a single agent, say i^*. In phase 2 (the second **while** loop in Algorithm 1), we continue to ask the worst-off agent to pick items in rounds. If the agent is not i^*, she can get a triplet in P. If the agent is i^*, we only allocate one vertex to her, which can always preserve EF1. To this end, we reserve a triplet in P and allocate one of the three vertices to i^*. The remaining vertices are temporarily stored in a new pool Q that cannot be given to any other agents except i^* in the second phase. In the later rounds, when i^* becomes the worst-off agent again, we continue to allocate one vertex in Q to her, and if Q becomes empty, we repeat the reservation process. Finally, when P becomes empty, we can use the ECE algorithm to allocate the remaining items.

There are two more issues that need to be addressed in order to guarantee high social welfare. First, in the partition (v_1, v_2, v_3), (v_4, v_5, v_6), etc., the broken edges such as (v_3, v_4) and (v_6, v_7) may bring very high utility to the agents and thus the sacrificed social welfare can be unbounded. Hence, instead of using one prefixed partition, we consider three shifted ones:

$$\begin{aligned} M_1 &= \{(v_1, v_2, v_3), (v_4, v_5, v_6), (v_7, v_8, v_9), \cdots\} \\ M_2 &= \{(v_1), (v_2, v_3, v_4), (v_5, v_6, v_7), \cdots\} \\ M_3 &= \{(v_1, v_2), (v_3, v_4, v_5), (v_6, v_7, v_8), \cdots\} \end{aligned} \tag{1}$$

The best of these three partitions ensures that we can preserve at least $2/3$ fraction of the optimal social welfare in the corresponding triplets. The second issue is that, we may break one triplet (say Θ^* from which the vertices in Q are selected) in phase 2, and this triplet may bring high utility to the agents. Hence, before moving to the ECE algorithm, we first check whether Θ^* or i^*'s partial allocation before she breaks Θ^* brings her higher utility. If Θ^* is better, we exchange i^*'s entire allocation by Θ^*. Attentive readers may note that after this exchange, the allocation may not be EF1 any more. Note that other agents do not envy agent i^* by more than one item, since Θ^* contains only one triplet, which can be destroyed by eliminating the middle vertex. However, agent i^* might envy others by more than one item. This is true, but fortunately, the later ECE algorithm can always compensate i^*'s loss and ensure EF1 for the final allocation.

Combining these two tricks with phases 1 and 2, we finally obtain the following result.

Theorem 4.1. *For any instance $\mathcal{I}$ with n homogeneous agents, Algorithm 1 returns in polynomial time an EF1 allocation with at least $2/3 - 2/(3(n+1))$ fraction of the optimal social welfare.*

Before proving Theorem 4.1, we first present several useful lemmas. For each M_l, $l = 1, 2, 3$, let $\mathrm{sw}(M_l)$ be the optimal social welfare by allocating the triples to the agents without EF1 constraints, disregarding all the utilities on the edges connecting different triples. We have the following lemma, which is proved in the appendix.

Lemma 4.2. *For any instance $\mathcal{I}$,*

$$\max\{\mathrm{sw}(M_1), \mathrm{sw}(M_2), \mathrm{sw}(M_3)\} \geq \frac{2}{3}\mathrm{sw}^*(\mathcal{I})$$

By Lemma 4.2, to ensure a good approximation of $\mathrm{sw}^*(\mathcal{I})$, it suffices to approximate the optimal social welfare of the best partition in M_1, M_2, M_3.

Lemma 4.3. *During the execution of Algorithm 1, all triplets in P are adjacent to the vertices assigned to the same agent. If $M \neq \emptyset$ or $P \neq \emptyset$, the partial allocation maintains EF1.*

Proof. Note that a triplet is put into the set P only when $M \neq \emptyset$. As shown in Algorithm 1, in each round of the first **while** loop (Step 8-13), an agent with the smallest utility is selected (say agent i). We compare agent i with the agent (say i^*) who receives a triplet in M in previous rounds. If they are the same agent, i.e., $i = i^*$, the leftmost triplet in M is put into P and the right contiguous triplet, if it exists, is allocated to the agent $i = i^*$. Therefore, all triplets within P connect the vertices assigned to the same agent. In what follows, we prove that during the execution of two **while** loops (Step 8-13 and Step 16-21, respectively), the partial allocation guarantees EF1 fairness. We begin with the analysis of the first **while** loop, which we move into only when $M \neq \emptyset$. Observe that in such case, we traverse the path. There are three cases:

- Case 1(a). $i \neq i^*$ and $P \neq \emptyset$;
- Case 1(b). $i \neq i^*$ and $P = \emptyset$;
- Case 1(c). $i = i^*$.

Consider Case 1(a), where a triplet $\Theta \in P$ is picked by agent i. Recall that all the triplets within P connect the vertices assigned to agent $i^* \neq i$. Therefore the allocation of a triplet $\Theta \in P$ can be regarded as allocating an imaginary item. Combining with the fact that i is the agent with the smallest utility before allocating the triplet Θ, allocating Θ still guarantees EF1 fairness. The similar reasoning can also be applied to Case 1(b), since the left contiguous triplet of Θ is allocated to agent $i^* \neq i$ and thus Θ can still be regarded as a single imaginary item. For Case 1(c), the leftmost triplets within M is temporarily put into set P so that the next triplet allocated to agent i connects no vertices of agent i's previous bundle. Hence, the next triplet within M is allocated to agent i, which can be regarded as an imaginary item and therefore does not break EF1 requirement. Next, we analyze the case of $M = \emptyset$ and $P \neq \emptyset$, for which we move into the second **while** loop. We discuss the following three cases:

- Case 2(a). $i \neq i^*$;
- Case 2(b). $i = i^*$ and $Q \neq \emptyset$;
- Case 2(c). $i = i^*$ and $Q = \emptyset$.

We first consider Case 2(a). Based on the previous analysis that all triplets within set P connects agent i^*'s assigned vertices, allocating a triplet within P to agent $i \neq i^*$ can be regarded as allocating a single item. Combining with the fact that agent i is the one with the smallest utility, we know that such allocation maintains EF1. Observe that in Case 2(b) a single item is allocated and in Case 2(c) no item is allocated to any agent, both of which maintains EF1 fairness. We complete the proof. □

Now, we are ready for the proof of Theorem 4.1.

Proof of Theorem 4.1. Recall that Lemma 4.3 completes the proof of maintaining EF1 fairness during the execution of the two **while** loops (Step 8-13 and Step 16-21, respectively). Moveover, although the bundle-exchanging procedure in Step 24 might break EF1 requirement, the later ECE procedure can compensate the welfare loss in the previous bundle-exchanging procedure and guarantee EF1 fairness for the final allocation. Next, we prove the social welfare guarantee. Observe that when we move out of the two **while** loops in Algorithm 1, there is at most one triplet (say Θ^*) being destroyed, i.e., part of vertices of Θ^* are allocated to an agent (say i^*) with the smallest utility and other vertices are temporarily stored in set Q. Consider the allocation $\mathbf{X}' = (X'_1, \ldots, X_n)$ before agent i^* picks the vertices from Θ^*. By Algorithm 1, we know that agent i^* is the worst-off one. Hence,

$$\begin{aligned} u(X'_{i^*}) &\leq \frac{1}{n}(u(X'_{i^*}) + \sum_{i \in N, i \neq i^*} u(X'_i)) \\ &\leq \frac{1}{n}(u(X'_{i^*}) + \sum_{i \in N, i \neq i^*} u(X_i)) \\ &= \frac{1}{n}(\sum_{\Theta \in M} u(V(\Theta)) - u(V(\Theta^*))) \\ &= \frac{1}{n}(\mathrm{sw}(M) - u(V(\Theta^*))), \end{aligned}$$

where the second inequality holds because $u(X'_i) \leq u(X_i), i \in N$. Recall that when we move out of the second **while** loop, we compare Θ with agent i^*'s assigned bundle X'_i before she breaks Θ^*. If Θ^* brings her larger utility, we exchange Θ^* with X'_i. Later we proceed with the ECE algorithm running on the remaining items which ensures a final EF1 allocation. Note that the ECE algorithm might not increase any utility. Therefore, the welfare loss would be no more than the utility of the smaller bundle within $\{\Theta^*, X'_{i^*}\}$. If $u(V(\Theta^*)) \geq u(X'_{i^*})$, the welfare loss is $u(X'_{i^*})$. We have

$$\begin{aligned} u(X_{i^*}) &\leq \min\{u(V(\Theta^*)), \frac{1}{n}(\mathrm{sw}(M) - u(V(\Theta^*)))\} \\ &\leq \frac{\mathrm{sw}(M)}{n+1}, \end{aligned}$$

where the equation of the second inequality holds only when $u(V(\Theta^*)) = \frac{\mathrm{sw}(M)}{n+1}$. Next, we deal with the case of $u(V(\Theta^*)) \leq u(X'_{i^*})$, where the welfare loss is $u(V(\Theta^*))$. Thus,

$$u(V(\Theta^*)) \leq u(X'_{i^*}) \leq \frac{1}{n}(\mathrm{sw}(M) - u(V(\Theta^*))),$$

Furthermore, we derive

$$u(V(\Theta^*)) \leq \frac{\mathrm{sw}(M)}{n+1}.$$

Based on the analysis above, we know that the welfare loss is restricted to be less than $\frac{\mathrm{sw}(M)}{n+1}$. The total welfare is

$$\sum_{i \in N} u(X_i) \geq \mathrm{sw}(M) - \frac{\mathrm{sw}(M)}{n+1} \geq \frac{2}{3}(1 - \frac{1}{n+1})\mathrm{sw}^*,$$

where the second inequality holds because $\mathrm{sw}(M) \geq (2/3)\mathrm{sw}^*$, proved in Lemma 4.2. Since in each round of the two **while** loops, exactly one triple or one item is allocated, and the envy-cycle elimination completes in $O(|V|)$ time, the algorithm runs in $O(|V|)$ time. We complete the proof of the theorem. □

The ratio of $2/3$ is actually the best possible guarantee even if the valuations are unary and there are only two agents. Consider a simple instance of allocating a path with four vertices and both agents have value 1 on all three edges. The optimal social welfare is 3 by allocating the entire path to one agent, but this allocation is not EF1. Thus any EF1 allocation will destroy at least one edge resulting in a social welfare of at most 2. In the appendix, we show

that when there are only two agents, the approximation ratio of 2/3 can be achieved.

Theorem 4.4. *For any instance $\mathcal{I}$ with two homogeneous agents, we can compute an EF1 allocation with at least 2/3 fraction of the optimal social welfare in polynomial time.*

5 IMPROVED APPROXIMATION FOR TWO HETEROGENEOUS AGENTS

In this section, we consider the case of two heterogeneous agents, and present a variant of the algorithm of Oh et al. [22] that achieves the best possible approximation ratio of the optimal social welfare.

To ensure a good approximation, we first compute an allocation $\mathbf{X} = (X_1, X_2)$ that maximizes the social welfare. By Lemma 2.2, such an allocation can be found in polynomial time. If $\mathbf{X}$ is already EF1, then we are done. Otherwise, without loss of generality, assume $u_1(X_1) < u_2(X_2)$. In the following, we show how to move vertices from X_2 to X_1 so that agent 1's utility can only increase and agent 2's utility cannot decrease by more than a half, and thus the remaining social welfare will be at least half of the optimal solution. First, we find an edge $e = (v_1, v_2)$ in X_2 whose removal divides X_2 into two bundles, the left bundle X_L and the right bundle X_R, such that $\max\{u_2(X_L), u_2(X_R)\} < 1/2 \cdot u_2(X_2)$. This can be done by checking the edges one by one starting from the leftmost and the first edge which makes the total weight be greater than $1/2 \cdot u_2(X_2)$ suffices. Next, we choose one of X_L and X_R whichever brings higher utility to agent 1, say X_L, and move vertices from X_L (and thus also from X_2) to X_1 one by one, until (1) agent 1 does not envy agent 2 by more than one vertex or (2) the two agents envy each other. If (1) happens, we can stop the algorithm, and if (2) happens, we can simply exchange their bundles. As shown by the hard instance in Theorem 3.1, this simple algorithm is actually the best possible.

Algorithm 2 Approximation Algorithm for Two Heterogeneous Agents

Input: Instance $\mathcal{I} = (G, N, \mathbf{w})$ with $G = (V, E)$.
Output: Allocation $\mathbf{X} = (X_1, X_2)$.

1: Compute a welfare maximizing allocation (M_1, M_2) and assume $u_1(M_1) \le u_2(M_2)$.
2: Initialize $X_i \leftarrow M_i, i \in \{1, 2\}$.
3: Partition X_2 as $X_L \cup X_R \cup \{v_1, v_2\}$ such that $\max\{u_2(X_L), u_2(X_R)\} < 1/2 \cdot u_2(X_2)$. Without loss of generality, suppose $u_1(X_R) \le u_1(X_L)$.
4: **while** agent 1 envies agent 2 by more than one item **do**
5: Let v be a vertex in X_L, and set $X_1 \leftarrow X_1 \cup \{v\}, X_L \leftarrow X_L \setminus \{v\}$, $X_2 \leftarrow X_2 \setminus \{v\}$.
6: **if** agent 2 envies agent 1 **then**
7: Exchange two agents' bundles, i.e., $X_2 \rightleftharpoons X_1$.
8: **end if**
9: **end while**
10: Output allocation $\mathbf{X} = (X_1, X_2)$.

The formal algorithm is shown in Algorithm 2, and we prove Theorem 5.1 in the appendix.

Theorem 5.1. *For any instance $\mathcal{I}$ with two heterogeneous agents, Algorithm 2 returns an EF1 allocation with at least 1/2 of the optimal social welfare in polynomial time.*

In the appendix, we also prove that, when the two agents' weights are binary, we can refine Algorithm 2 and achieve 2/3 approximation, as shown in Theorem 5.2. Recall the hard instance below Theorem 4.1, 2/3 is the best possible approximation ratio.

Theorem 5.2. *For any instance $\mathcal{I}$ with two heterogeneous agents whose weight functions are binary, an EF1 allocation with at least 2/3 of the maximum social welfare can be found in polynomial time.*

6 IMPROVED APPROXIMATION FOR HETEROGENEOUS BINARY WEIGHT FUNCTIONS

Finally, we study the case with arbitrary number of agents whose weight functions are heterogeneous and binary.

Let us start with the following instance with n agents. We have n identical gadgets, where each gadget $k = 1, \ldots, n$ contains four vertices $v_{k1}, v_{k2}, v_{k3}, v_{k4}$. Agent 1 has weight 1 on the three edges (v_{k1}, v_{k2}), (v_{k2}, v_{k3}), (v_{k3}, v_{k4}) and all the other agents $i \ge 2$ has weight 1 on (v_{k2}, v_{k3}) but 0 on (v_{k1}, v_{k2}) and (v_{k3}, v_{k4}). All gadgets are connected as a path, as shown in Figure 3, and all n agents have weight 0 on the connecting edges $(v_{k4}, v_{k+1,1})$ for $k = 1, \ldots, n-1$. It is easy to see that the optimal social welfare is achieved by allocating the whole graph to agent 1 and $\mathrm{sw}^* = 3n$. However, in any EF1 allocation, agent 1 cannot have utility greater than 3. Suppose otherwise, then agent 1 must receive edges from at least two different gadgets, and no other agent $i \ge 2$ can obtain positive utilities from these two gadgets. Since there are at most $n-2$ gadgets that have not been allocated to agent 1, there must be some agent $i \ge 2$ who envies agent 1 by more than one item, and thus the allocation is not EF1. Combining with the fact that the maximum partial social welfare of agents $2, \ldots, n$ is no greater than $n-1$, the maximum social welfare that can be obtained by EF1 allocations is $n+2$. Thus, when n goes to infinity, we conclude that no EF1 allocation can guarantee strictly more than 1/3 fraction of the optimal social welfare. Remaining unassigned vertices are

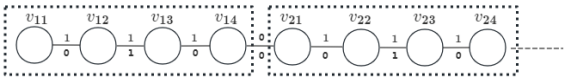

Figure 3: Hard instance for binary weight functions.

allocated by the envy-cycle elimination procedure [20], which still maintains the EF1 fairness and does not decrease any social welfare. Now, we provide the intuition behind Algorithm 3, which processes the vertices one by one. As illustrated in Figure 4, in each round we try to find the leftmost remaining edge $e_k = (v_k, v_{k+1})$ with weight 1 for at least one agent. Consider a set S of agents who have weight 1 on edge e_k. We allocate the two endpoints $\{v_k, v_{k+1}\}$ to one agent $i \in S$ with the smallest bundle. In the next round, if $\{v_{k+2}, v_{k+3}\}$ is the next pair to be assigned, a tricky case might happen which cannot guarantee EF1 fairness. Let S' be the set of agents with weight 1 on edge (v_{k+2}, v_{k+3}). If we try to allocate $\{v_{k+2}, v_{k+3}\}$ to

Algorithm 3 Approximation Algorithm for Binary Weights

Input: Instance $\mathcal{I} = (G, N, \mathbf{w})$ with $G = (V, E)$.
Output: Allocation $\mathbf{X} = (X_1, \ldots, X_n)$.

Initialize $X_i \leftarrow \emptyset$ for all $i \in N$.
Let P be the set of unallocated vertices. Initialize $P \leftarrow V$.
Initialize $k \leftarrow 1$.
while $k \leq |V| - 1$ **do**
  Let S be the set of agents who have weight 1 on the k-th edge $e_k = (v_k, v_{k+1})$.
  if $S = \emptyset$ **then**
    $k \leftarrow k + 1$.
  else
    Let $i^* \in \arg\min_{i \in S} |X_i|$ be an agent who has smallest bundle in S.
    Set $X_{i^*} \leftarrow X_{i^*} \cup \{v_k, v_{k+1}\}$, $P \leftarrow P \setminus \{v_k, v_{k+1}\}$, $k \leftarrow k + 3$.
  end if
end while
Execute ECE on P.
Return Allocation $(X_1, \ldots, X_n)$.

an agent $i' \in S'$ (weights are above the edges) with the smallest bundle. There might exist an agent $i \in N \setminus S'$ (weights are below the edges) who envies agent i' up to one item before allocating $\{v_{k+2}, v_{k+3}\}$ and has weight 1 on edge (v_{k+1}, v_{k+2}). Therefore, the EF1 fairness is broken after allocating $\{v_{k+2}, v_{k+3}\}$. To circumvent this obstacle, we temporarily abandon the node v_{k+2} and consider edge (v_{k+3}, v_{k+4}) so that each allocated pairs can be regarded as an imaginary item. If no agent has weight 1 on edge (v_{k+3}, v_{k+4}), we skip node v_{k+3} and consider the next edge (v_{k+4}, v_{k+5}). Otherwise, we assign $\{v_{k+3}, v_{k+4}\}$ to an agent to increase the total welfare by 1. Since the agent receiving $\{v_{k+3}, v_{k+4}\}$ is not envied by any other agent with weight 1 on this edge and the allocation has no effect on those agents with weight 0, such allocation still guarantees EF1 fairness. Repeating the process above until the whole path has been traversed, we obtain a partial EF1 allocation. Whenever we allocate a pair of vertices to increase the total welfare by 1, there are at most two consecutive edges being abandoned. Furthermore, the ECE procedure does not decrease any social welfare. Therefore, we retain at least 1/3 of the optimal social welfare. Thus, we have Theorem 6.1, proved in the appendix.

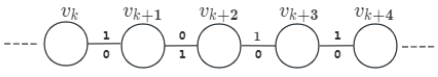

Figure 4: A path is distributed among n agents with binary weights.

Theorem 6.1. *For any instance $\mathcal{I}$ with binary weights, Algorithm 3 returns an EF1 allocation with social welfare at least $1/3 \cdot \mathrm{sw}^*(\mathcal{I})$ in polynomial time.*

7 EXTENSION: GENERAL GRAPHS

In this section, we extend our results to more general settings – *trees* and *general graphs*. In particular, the resources are modelled as the vertices in a graph and the agents may have different weights on the edges. The utility of an agent is represented by the total weight of the edges within the induced subgraph of her received vertices. Due to space limit, we defer the detailed discussion to the appendix.

In the following, we briefly summarize our results. Without considering fairness requirements, we first study the computational complexity of computing a welfare-maximizing allocation:

- A welfare-maximizing allocation for trees can be computed efficiently by a similar algorithm used for path;
- In sharp contrast, it is NP-hard to compute a welfare-maximizing allocation for the general graph.

Regarding the social welfare guarantee of EF1 allocations, our results are summarized in Table 2.

	heterogeneous agents		homogeneous agents
	binary weights	arbitrary weights	arbitrary weights
trees	$\Theta(1/n)$		$1/2$
general graphs	$1/(4\Delta - 2)$	$2/V(V-1)$	

Table 2: A summary of extension, where V is the number of vertices in the graph and Δ is the maximum vertex degree.

8 CONCLUSION

In this paper, we study the discrete cake-cutting problem, where cutting destroys the pieces and thus causes utility loss. We design efficient algorithms to find EF1 allocations with guaranteed social welfare that are (approximately) optimal for various settings. In the appendix, we show how to extend our results to more general settings where the underlying graph can be arbitrary and not restricted to the path. For trees, as an example, most positive results carry through, such as the efficient computation of the optimal social welfare and the approximation algorithms for the settings we have considered. However, for more general classes of graphs, the news is less positive. Notably, it is NP-hard to find allocations maximizing the social welfare. Also, to guarantee EF1 fairness, the welfare loss could be very high even in the special settings such as binary weight functions.

There are many interesting future directions. An immediate one is to consider other fairness solution concepts, such as MMS, PROP, EFX, etc. Another direction would be analyzing other preference representations, such as chores, mixture of goods and chores, and mixture of divisible and indivisible items. We can also study the constrained settings; for example, computing the price of fairness when each agent's allocation is required to be connected.

ACKNOWLEDGMENTS

Research of Zheng Chen and Guochuan Zhang was supported by Science and Technology Innovation 2030 - "The Next Generation of Artificial Intelligence" Major Project (2018AAA0100902). Research of Bo Li was supported by the National Natural Science Foundation of China (No. 62102333) and Hong Kong SAR Research Grants Council (No. PolyU 15224823).

REFERENCES

[1] Noga Alon. 1987. Splitting necklaces. *Advances in Mathematics* (1987).

[2] Haris Aziz and Simon Mackenzie. 2016. A Discrete and Bounded Envy-Free Cake Cutting Protocol for Any Number of Agents. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS*. IEEE Computer Society, 416–427.

[3] Haris Aziz and Simon Mackenzie. 2016. A discrete and bounded envy-free cake cutting protocol for four agents. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC*. ACM, 454–464.

[4] Siddharth Barman, Umang Bhaskar, and Nisarg Shah. 2020. Optimal Bounds on the Price of Fairness for Indivisible Goods. In *WINE (Lecture Notes in Computer Science, Vol. 12495)*. Springer, 356–369.

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

[6] Xiaohui Bei, Xinhang Lu, Pasin Manurangsi, and Warut Suksompong. 2021. The Price of Fairness for Indivisible Goods. *Theory Comput. Syst.* 65, 7 (2021), 1069–1093.

[7] Dimitris Bertsimas, Vivek F. Farias, and Nikolaos Trichakis. 2011. The Price of Fairness. *Operations Research* 59, 1 (2011), 17–31.

[8] Vittorio Bilò, Ioannis Caragiannis, Michele Flammini, Ayumi Igarashi, and Gianpiero Monaco. 2022. Almost envy-free allocations with connected bundles. *Games Econ. Behav.* 131 (2022), 197–221.

[9] Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, and Dominik Peters. 2017. Fair Division of a Graph. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*. 135–141.

[10] Sylvain Bouveret, Katarína Cechlárová, and Julien Lesca. 2019. Chore division on a graph. *Auton. Agents Multi Agent Syst.* 33, 5 (2019), 540–563.

[11] Steven J. Brams and Alan D. Taylor. 1995. An Envy-Free Cake Division Protocol. *Amer. Math. Monthly* 102, 1 (1995).

[12] I Bárány and V. S. Grinberg. 2015. Block partitions of sequences. *Israel Journal of Mathematics* (2015).

[13] Ioannis Caragiannis, Christos Kaklamanis, Panagiotis Kanellopoulos, and Maria Kyropoulou. 2012. The Efficiency of Fair Division. *Theory Comput. Syst.* 50, 4 (2012), 589–610.

[14] Francis Edward Su. 1999. Rental Harmony: Sperner's Lemma in Fair Division. *Amer. Math. Monthly* (1999).

[15] D. K. Foley. 1967. Resource Allocation and the Public Sector. *Yale Econ. Essays* 45-98 (1967).

[16] Gianluigi Greco and Francesco Scarcello. 2020. The Complexity of Computing Maximin Share Allocations on Graphs. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*. AAAI Press, 2006–2013.

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

[18] Ayumi Igarashi and Dominik Peters. 2019. Pareto-Optimal Allocation of Indivisible Goods with Connectivity Constraints. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI*. AAAI Press, 2045–2052.

[19] Ayumi Igarashi and William S. Zwicker. 2023. Fair division of graphs and of tangled cakes. *Mathematical Programming* (2023).

[20] R. Lipton, E. Markakis, E. Mossel, and A. Saberi. 2004. On approximately fair allocations of indivisible goods. In *EC*. 125–131.

[21] Javier Marenco and Tomás Tetzlaff. 2014. Envy-free division of discrete cakes. *Discret. Appl. Math.* 164 (2014), 527–531.

[22] Hoon Oh, Ariel D. Procaccia, and Warut Suksompong. 2021. Fairly Allocating Many Goods with Few Queries. *SIAM J. Discret. Math.* 35, 2 (2021), 788–813.

[23] Ariel D. Procaccia. 2009. Thou Shalt Covet Thy Neighbor's Cake. In *Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence, IJCAI*. 239–244.

[24] Hugo Steinhaus. 1948. The Problem of Fair Division. *Econometrica* 16, 1 (1948).

[25] Hugo Steinhaus. 1949. Sur la division pragmatique. *Econometrica* 17 (Supplement) (1949), 315–319.

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

[27] Miroslaw Truszczynski and Zbigniew Lonc. 2020. Maximin Share Allocations on Cycles. *J. Artif. Intell. Res.* 69 (2020), 613–655.