

Approximate Envy-Freeness in Graphical Cake Cutting

Sheung Man Yuen and Warut Suksompong

National University of Singapore

{yuens,warut}@comp.nus.edu.sg

Abstract

We study the problem of fairly allocating a divisible resource in the form of a graph, also known as graphical cake cutting. Unlike for the canonical interval cake, a connected envy-free allocation is not guaranteed to exist for a graphical cake. We focus on the existence and computation of connected allocations with low envy. For general graphs, we show that there is always a $1/2$ -additive-envy-free allocation and, if the agents' valuations are identical, a $(2 + \epsilon)$ -multiplicative-envy-free allocation for any $\epsilon > 0$. In the case of star graphs, we obtain a multiplicative factor of $3 + \epsilon$ for arbitrary valuations and 2 for identical valuations. We also derive guarantees when each agent can receive more than one connected piece. All of our results come with efficient algorithms for computing the respective allocations.

1 Introduction

Cake cutting refers to the classic problem of fairly allocating a divisible resource such as land or advertising spaces—playfully modeled as a “cake”—among agents who may have different values for different parts of the resource [Robertson and Webb, 1998; Procaccia, 2013]. The most common fairness criteria in this literature are *proportionality* and *envy-freeness*. Proportionality demands that if there are n agents among whom the cake is divided, then every agent should receive at least $1/n$ of her value for the entire cake. Envy-freeness, on the other hand, requires that no agent would rather have another agent's piece of cake than her own. Early work in cake cutting established that a proportional and envy-free allocation that assigns to each agent a connected piece of cake always exists, regardless of the number of agents or their valuations over the cake [Dubins and Spanier, 1961; Stromquist, 1980; Su, 1999].

Although existence results such as the aforementioned guarantees indicate that a high level of fairness can be achieved in cake cutting, they rely on a typical assumption in the literature that the cake is represented by an interval. This representation is appropriate when the resource corresponds to machine processing time or a single road, but becomes insufficient when one wishes to divide more complex

resources such as networks. For example, one may wish to divide road networks, railway networks, or power cable networks among different companies for the purpose of construction or maintenance. In light of this observation, Bei and Suksompong [2021] introduced a more general model called *graphical cake cutting*, wherein the cake can be represented by any connected graph. With a graphical cake, a connected proportional allocation may no longer exist—see Figure 1 (left). Nevertheless, these authors showed that more than half of the proportionality guarantee can be retained: any graphical cake admits a connected allocation such that every agent receives at least $1/(2n - 1)$ of her entire value.

The result of Bei and Suksompong [2021] demonstrates that approximate proportionality is attainable in graphical cake cutting. However, the allocation that their algorithm produces may lead to high *envy* between the agents. In particular, while each agent i is guaranteed $1/(2n - 1)$ of her value, it is possible that the algorithm assigns the remaining $(2n - 2)/(2n - 1)$ of the value to another agent j from i 's perspective, so that i envies j by almost the entire value of the cake (for large n) when measured additively, and by a factor linear in n when measured multiplicatively. Note that envy-freeness is a much more stringent benchmark than proportionality—for instance, although there exists a simple protocol for computing a connected proportional allocation of an interval cake [Dubins and Spanier, 1961], no finite protocol can compute a connected envy-free allocation of it [Stromquist, 2008], and even without the connectivity requirement, the only known envy-free protocol requires an enormous number of queries [Aziz and Mackenzie, 2016]. The goal of our work is to investigate the existence of connected allocations of a graphical cake with low envy, as well as to design algorithms for computing such allocations.

1.1 Our Results

We assume that the cake is represented by the edges of a connected graph, and each edge can be subdivided into segments to be allocated to different agents. Each agent is to receive a connected piece, though we will also briefly explore relaxations of this constraint towards the end. The whole cake must be allocated, and each agent's value for it is additive and normalized to 1; the value of each agent does not need to be uniform within each edge or across different edges. Following Bei and Suksompong [2021],

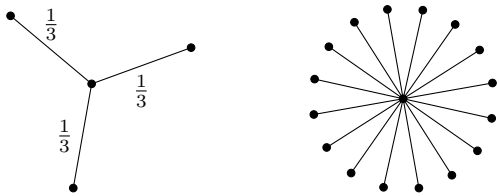


Figure 1: (Left) A star graph with three edges of equal length. Two agents with identical valuations distributed uniformly over the three edges cannot each receive a connected piece worth at least $1/2$ of the whole cake at the same time. (Right) A star graph with many edges to be divided between two agents. If sharing of vertices is disallowed, then the agent who does not receive the center vertex will be restricted to at most one edge, and will incur envy equal to almost the value of the entire cake.

we also assume that each vertex can be shared by multiple agents.¹ We consider both additive envy—for $\alpha \in [0, 1]$, an allocation is α -additive-EF if no agent envies another agent by an amount of more than α —and multiplicative envy—for $\alpha \geq 1$, an allocation is α -EF if no agent envies another agent by a factor of more than α . An α -EF allocation is also $(\frac{\alpha-1}{\alpha+1})$ -additive-EF; we refer to Proposition 1 for details.

In Section 3, we consider agents with (possibly) non-identical valuations. We show that for any graph, there exists a $1/2$ -additive-EF allocation, and such an allocation can be computed by iteratively allocating to each agent a share that other agents do not value too highly. If the graph is a star, we present an algorithm that, for any $\epsilon > 0$, finds a $(3 + \epsilon)$ -EF allocation (which is therefore nearly $1/2$ -additive-EF as well) by allowing agents to repeatedly relinquish their current share for a higher-value share, and allocating the remaining shares by following certain rules. Our two algorithms generalize ideas from algorithms for the interval cake by Goldberg *et al.* [2020] and Arunachaleswaran *et al.* [2019], respectively. We remark here that star graphs are of particular interest in graphical cake cutting because they constitute perhaps the most intuitive generalization of the well-studied interval cake, and therefore provide a natural platform for attempts to extend techniques and results from the interval-cake setting.²

Next, in Section 4, we demonstrate how the bounds for non-identical valuations can be improved in the case of identical valuations; this case captures scenarios in which there is an objective valuation among agents.³ For arbitrary graphs, we devise an algorithm that computes a $(2 + \epsilon)$ -EF allocation (which is therefore nearly $1/3$ -additive-EF). Our algorithm is inspired by the work of Chu *et al.* [2010] on partitioning edges of a graph (see Section 1.2), and involves repeatedly adjusting the shares along a path from the minimum share to the maximum share so that

¹Without this assumption, one cannot obtain nontrivial envy-free guarantees—see the caption of Figure 1 (right) for a brief discussion.

²Star graphs (and path graphs) are also often studied in the context of indivisible items (see Section 1.2). In graphical cake cutting, all path graphs are equivalent to the classic interval cake, which is why the role of star graphs is further highlighted.

³We discuss further motivation for investigating this case at the beginning of Section 4.

	General graphs	Star graphs
Non-identical valuations	$1/2$ -additive-EF (Thm. 3)	$(3 + \epsilon)$ -EF (Thm. 4)
Identical valuations	$(2 + \epsilon)$ -EF (Thm. 6)	2-EF (Thm. 7)

Table 1: Summary of results in Sections 3 and 4.

the shares become more balanced in value. For star graphs, we provide a simpler algorithm that returns a 2-EF allocation using a bag-filling idea. As we discuss after Proposition 1, an approximate proportionality result of Bei and Suksompong [2021] implies that both of our guarantees in this section are (essentially) tight.

Finally, in Section 5, we explore the fairness guarantees when each agent can receive more than one connected piece. We introduce the notion of *path similarity number* to discuss the relationship between connected interval cake cutting and (non-connected) graphical cake cutting.

Our results in Sections 3 and 4 are summarized in Table 1, and all proofs can be found in the full version of our paper [Yuen and Suksompong, 2023]. All of our algorithms can be implemented in the standard cake-cutting model of Robertson and Webb [1998] in time polynomial in n , the size of the graph, and, if applicable, $1/\epsilon$.

1.2 Further Related Work

Cake cutting is a topic of constant interest for researchers in mathematics, economics, and computer science alike. For an overview of its intriguing history, we refer to the books by Brams and Taylor [1996] and Robertson and Webb [1998], as well as the book chapter by Procaccia [2016].

The cake-cutting literature traditionally assumes that the cake is given by an interval, and connectivity of the cake allocation is often desired in order to avoid giving agents a “union of crumbs” [Stromquist, 1980; Stromquist, 2008; Su, 1999; Bei *et al.*, 2012; Cechlárová *et al.*, 2013; Arunachaleswaran *et al.*, 2019; Goldberg *et al.*, 2020; Barman and Kulkarni, 2022].⁴ Besides Bei and Suksompong [2021], a few authors have recently addressed the division of a graphical cake. Igarashi and Zwicker [2021] focused on envy-freeness but made the crucial assumption that vertices cannot be shared between agents—as discussed in the caption of Figure 1 (right), with their assumption, one cannot obtain nontrivial guarantees even for star graphs and identical valuations. Deligkas *et al.* [2022] explored the complexity of deciding whether an envy-free allocation exists for a given instance (and, if so, finding one), both when vertices can and cannot be shared, but did not consider approximate envy-freeness. Elkind *et al.* [2021] investigated another fairness notion called maximin share fairness in graphical cake cutting.

For additional discussion of related work, please refer to the full version of our paper [Yuen and Suksompong, 2023].

⁴While connectivity is the most frequently studied constraint in cake cutting, other constraints, such as geometric and separation constraints, have also been explored [Suksompong, 2021].

2 Preliminaries

Let $G = (V, E)$ be a connected undirected graph representing the cake. Each edge $e \in E$, isomorphic to the interval $[0, 1]$ of the real numbers, is denoted by $e = [v_1, v_2] = [v_2, v_1]$ where $v_1, v_2 \in V$ are the endpoints of edge e . If x_1 and x_2 are points *on* an edge, then the segment between them is denoted by $[x_1, x_2]$ or $[x_2, x_1]$ —we sometimes call it an *interval*. We shall restrict our attention to closed intervals only.

Two intervals of a cake G are considered *disjoint* if their intersection is a finite set of points. A *share* of a cake G is a finite union of pairwise disjoint (closed) intervals of G —where the intervals may belong to different edges—such that it is connected, i.e., for any two points in the share, there exists a path between the two points that only traverses the share.⁵ As with intervals, two shares of a cake G are considered *disjoint* if their intersection is a finite set of points.

Let $N = \{1, 2, \dots, n\}$ be a set of $n \geq 2$ agents. Each agent $i \in N$ has a *valuation function* (or *utility function*) μ_i , which maps each share of G to a nonnegative real number. Following the cake-cutting literature [Procaccia, 2016], we assume that each valuation function μ_i is

- *normalized*: $\mu_i(G) = 1$,
- *divisible*: for each interval $[x_1, x_2]$ and $\lambda \in [0, 1]$, there exists a point $y \in [x_1, x_2]$ such that $\mu_i([x_1, y]) = \lambda \cdot \mu_i([x_1, x_2])$, and
- *additive*: for any two disjoint intervals I_1 and I_2 , we have $\mu_i(I_1 \cup I_2) = \mu_i(I_1) + \mu_i(I_2)$.

An *instance* of graphical cake cutting consists of a graph G , a set of agents N , and valuation functions $(\mu_i)_{i \in N}$.

Given an instance, a *partial allocation* of G is a list $\mathcal{A} = (A_1, \dots, A_n)$ of pairwise disjoint shares of the cake, where A_i is agent i 's share—we say that A_i is *allocated* to agent i . A share is *unallocated* if its intersection with every agent's share is a finite set of points. An *allocation* of G is a partial allocation such that every point in the cake is allocated to some agent. For a parameter α , a partial allocation \mathcal{A} is

- α -*additive-EF* if $\mu_i(A_i) \geq \mu_i(A_j) - \alpha$ for all $i, j \in N$,
- α -*EF* if $\mu_i(A_i) \geq \mu_i(A_j)/\alpha$ for all $i, j \in N$,
- *envy-free* if \mathcal{A} is 1-EF (or, equivalently, 0-additive-EF),
- α -*proportional* if $\mu_i(A_i) \geq 1/(\alpha n)$ for all $i \in N$, and
- *proportional* if \mathcal{A} is 1-proportional.

In order for algorithms to access the cake valuations, we assume that they can make CUT and EVAL queries available in the standard model of Robertson and Webb [1998].

For completeness, we state the relationships between different fairness notions.

Proposition 1. *Let \mathcal{A} be an allocation for $n \geq 2$ agents, and let $\alpha \geq 1$.*

- *If \mathcal{A} is α -EF, then it is $(\alpha - \frac{\alpha-1}{n})$ -proportional.*
- *If \mathcal{A} is α -EF, then it is $(\frac{\alpha-1}{\alpha+1})$ -additive-EF.*

⁵For brevity, we use the term “share” for what is sometimes referred to as a “connected piece of cake”.

- *If \mathcal{A} is α -proportional, then it is $(1 - \frac{2}{\alpha n})$ -additive-EF.*

Bei and Suksompong [2021] demonstrated that for every $n \geq 2$, there exists an instance in which no allocation is α -proportional for any $\alpha < 2 - 1/n$, even for identical valuations and star graphs. By Proposition 1, one cannot obtain a better guarantee than 2-EF for such instances.

We now state a useful lemma about the existence of a share that has sufficiently high value for one agent and, at the same time, not exceedingly high value for other agents.

Lemma 2. *Let H be a connected subgraph of a graphical cake, and suppose that H is worth β_0 to some agent in a subset $N' \subseteq N$. Then, for any positive $\beta \leq \beta_0$ and any vertex r of H , there exists an algorithm, running in time polynomial in n and the size of H , that finds a partition of H into two (connected) shares such that the first share is worth at least β to some agent in N' and less than 2β to every agent in N' , and the second share contains the vertex r .*

Bei and Suksompong [2021, Lemma 4.9] made this claim for the special case where all agents have the same value for H and no vertex r is specified. We will use Lemma 2 as a subroutine in ITERATIVEDIVIDE (Algorithm 1) and BALANCEPATH (Algorithm 4); ITERATIVEDIVIDE considers the case where different agents may have different values for H , while BALANCEPATH requires the condition on the vertex r in order to maintain connectivity along the minimum-maximum path. We shall use $\text{DIVIDE}(H, N', \beta, r)$ to denote the ordered pair of the two corresponding shares as described in the lemma. The idea behind the proof of Lemma 2 is similar to that of the special case shown by Bei and Suksompong [2021]: we convert H into a tree rooted at the vertex r by removing cycles iteratively—keeping the edges and duplicating the vertices if necessary—then traverse the tree from r until a vertex v with a subtree of an appropriate size is reached, and finally identify some connected subgraph of the subtree as the first share while assigning the remaining portion as the second share. The full details, including the pseudocode, are given in the full version of our paper [Yuen and Suksompong, 2023].

3 Non-Identical Valuations

In this section, we allow agents to have different valuations. For arbitrary graphs, we present an algorithm that computes an approximately envy-free allocation of a graphical cake when measured additively. For the case where the graph is a star, we give an algorithm that finds an allocation wherein the envy is bounded by a multiplicative factor of roughly 3.

3.1 General Graphs

A priori, it is not even clear whether there exists a constant $\alpha < 1$ independent of n such that an α -additive-EF allocation always exists. We now describe the algorithm, ITERATIVEDIVIDE (Algorithm 1), which finds a $1/2$ -additive-EF allocation for arbitrary graphs and non-identical valuations, using ideas similar to the algorithm by Goldberg *et al.* [2020] for computing a $1/3$ -additive-EF allocation of an interval cake. Choose any arbitrary vertex r of G , and start with the entire graph G and all agents in contention. If there

Algorithm 1 ITERATIVEDIVIDE(G, N).

Input: Graph G , set of agents $N = \{1, \dots, n\}$.
Output: Allocation (A_1, \dots, A_n) .
Initialization: $r \leftarrow$ any vertex of G ; $H_n \leftarrow G$; $N' \leftarrow N$.

- 1: **for** $i = 1, \dots, n - 1$ **do**
- 2: $\beta_i \leftarrow 1/4$
- 3: **if** there exists $i' \in N'$ such that $\mu_{i'}(H_n) \geq \beta_i$ **then**
- 4: $(H_i, H_n) \leftarrow \text{DIVIDE}(H_n, N', \beta_i, r)$
- 5: $i^* \leftarrow$ any agent in N' who values H_i at least β_i
- 6: **else**
- 7: $H_i \leftarrow \emptyset$
- 8: $i^* \leftarrow$ any agent in N'
- 9: **end if**
- 10: $A_{i^*} \leftarrow H_i$
- 11: $N' \leftarrow N' \setminus \{i^*\}$
- 12: **end for**
- 13: $A_j \leftarrow H_n$, where j is the remaining agent in N'
- 14: **return** (A_1, \dots, A_n)

is only one agent remaining, allocate the remaining graph to that agent. If the remaining graph is worth less than $\beta = 1/4$ to every remaining agent,⁶ allocate an empty graph to any one of the remaining agents and remove this agent. Otherwise, apply the algorithm DIVIDE on the remaining graph and the remaining agents with threshold β . Allocate the first share to any agent who values that share at least β , and remove this agent along with her share. Repeat the procedure with the remaining graph until the whole graph is allocated. We claim that the resulting allocation is indeed 1/2-additive-EF.

Theorem 3. *Given an instance of graphical cake cutting, there exists an algorithm that computes a 1/2-additive-EF allocation in time polynomial in n and the size of G .*

While an additive envy of 1/2 can be seen as high, the left example of Figure 1 shows that an envy of 1/3 is inevitable. Moreover, even for an interval cake, the (roughly) 1/4-additive approximation of Barman and Kulkarni [2022] is the current best as far as polynomial-time computability is concerned.

Although ITERATIVEDIVIDE guarantees that the envy between each pair of agents is at most 1/2, it is possible that some agents receive an empty share from the algorithm. In the remainder of this paper, we present algorithms that find approximately envy-free allocations up to constant multiplicative factors for star graphs as well as for agents with identical valuations. Any such allocation ensures positive value for every agent and, by Proposition 1, is also approximately envy-free when measured additively.

3.2 Star Graphs

The case of star graphs presents a natural generalization of the canonical interval cake and, as can be seen in Figure 1, already highlights some of the challenges that graphical cake cutting poses. For this class of graphs, we devise an algorithm

⁶While the value of β is the same for all iterations here, we write β_i in the pseudocode because we will later consider a generalization in which β can be different for different iterations.

that, for any constant $\epsilon > 0$, computes a $(3 + \epsilon)$ -EF allocation in polynomial time. The algorithm consists of four phases. It starts with an empty partial allocation and finds a small star of “stubs” near the center vertex (Phase 1). It then repeatedly finds an unallocated share worth slightly more than some agent’s share, and allows that agent to relinquish her existing share for this new share—care must be taken to ensure that other agents do not have too much value for this new share (Phase 2). This new share could be a segment of an edge (Phase 2a) or a union of multiple complete edges (Phase 2b). This phase is repeated until there are no more unallocated shares suitable for agents to trade with. Finally, the unallocated shares are appended to the agents’ existing shares (Phases 3 and 4). See Figure 2 for an illustration of each phase. We remark that Phases 2a and 3 of our algorithm are adapted from the algorithm of Arunachaleswaran *et al.* [2019] for finding a $(2 + \epsilon)$ -EF allocation of an interval cake.

Let $G = (V, E)$ be a star graph centered at vertex v with $m \geq 2$ edges. Label the other vertices v_k and the edges $e_k = [v_k, v]$ for $k \in \{1, \dots, m\}$. Fix any $\epsilon \in (0, 1)$.

Phase 1: Preparation. Define $\epsilon' = \frac{\epsilon}{16nm}$. Initialize an empty partial allocation $\mathcal{A} = (A_1, \dots, A_n)$. For each edge e_k , find a point $x_k \in [v_k, v]$ such that the segment $[x_k, v]$ is worth at most ϵ'/m to all agents. Define $e_k^1 = [v_k, x_k]$ and $e_k^2 = [x_k, v]$ for $k \in \{1, \dots, m\}$, and let E^1 and E^2 be the sets containing all e_k^1 ’s and all e_k^2 ’s, respectively. Note that E^2 is worth at most ϵ' to every agent.

Phase 2: Increase agents’ shares incrementally. If there is a segment $e_k^1 \in E^1$ such that some unallocated interval within e_k^1 is worth at least $\mu_i(A_i) + \epsilon'$ to some agent i , **go to Phase 2a**. Otherwise, consider the segments in E^1 that are entirely unallocated. If the union of these segments is worth at least $\mu_i(A_i) + \epsilon'$ to some agent i , **go to Phase 2b**. Otherwise, Phase 2 ends; **go to Phase 3**.

- **Phase 2a: Allocate a subinterval of some e_k^1 .** Pick an unallocated interval $I \subseteq e_k^1$ that is worth at least $\mu_i(A_i) + \epsilon'$ to some agent i , and assume without loss of generality that it cannot be extended in either direction without overlapping an allocated share or e_k^2 . Suppose that $I = [a, b]$, where a is closer to v_k than b is. If $a = v_k$, find the point $z \in I$ closest to a such that $[a, z]$ is worth exactly $\mu_{i^*}(A_{i^*}) + \epsilon'$ to some agent i^* , and let $A_{i^*} = [a, z]$, i.e., agent i^* relinquishes her existing share for this new share. Else, $a \neq v_k$; find the point $z \in I$ closest to b such that $[z, b]$ is worth exactly $\mu_{i^*}(A_{i^*}) + \epsilon'$ to some agent i^* , and let $A_{i^*} = [z, b]$. **Repeat Phase 2**.
- **Phase 2b: Allocate multiple edges in E .** Let K_0 be the set of all indices k such that the entire segment e_k^1 is unallocated. Initialize $K = \emptyset$, and add the indices from K_0 to K one by one until $\{e_k^1 \mid k \in K\}$ is worth at least $\mu_{i^*}(A_{i^*}) + \epsilon'$ to some agent i^* . Let A_{i^*} be the union of e_k over all $k \in K$, i.e., agent i^* relinquishes her existing share for this new share. Note that this new share is connected by the center vertex v . **Repeat Phase 2**.

Phase 3: Append unallocated subintervals within e_k^1 . Let $N_1 \subseteq N$ consist of all agents who last received a subinterval of some e_k^1 via Phase 2a, and $N_2 \subseteq N$ consist of all agents

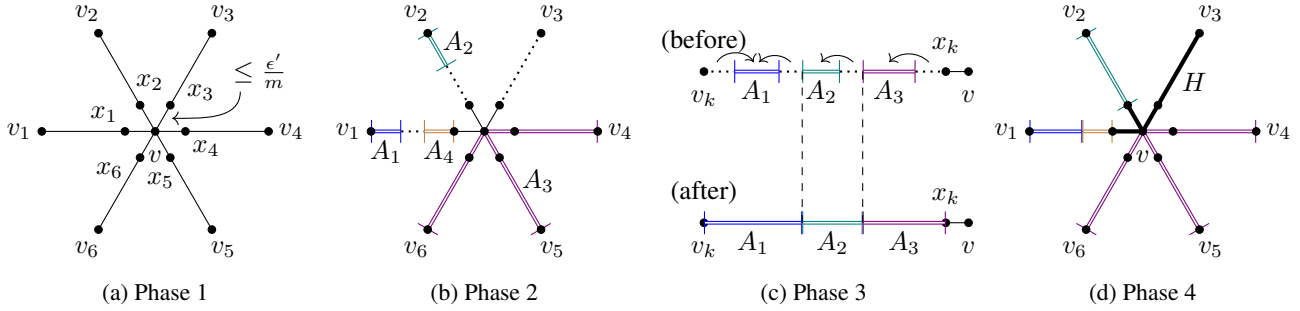


Figure 2: (a) The points x_k are found, where $[x_k, v]$ is worth at most ϵ'/m to every agent. (b) The unallocated intervals (dotted lines) are the ones to be considered in Phase 2a. (c) The unallocated intervals (dotted lines) are appended leftwards in v_k 's direction, except for the one containing v_k which is appended rightwards. (d) The remaining unallocated portion H (bold lines) is a share connected by v .

who last received two or more complete edges in E via Phase 2b (we will show later that, in fact, $N_1 \cup N_2 = N$). For each e_k^1 of which some agent from N_1 is allocated a subinterval, and for each unallocated interval $I = [a, b] \subseteq e_k^1$ that cannot be extended in either direction without overlapping an allocated share or E^2 , where a is closer to v_k than b is, append I to the share of the agent who is allocated the point a (i.e., append towards v_k 's direction). The only time this is not possible is when $a = v_k$, in which case we append to the share of the agent who is allocated the point b .

Phase 4: Append H . Consider the remaining unallocated portion H of the graph. Note that for each $k \in \{1, \dots, m\}$, we have $H \cap e_k = \{v\}$ or e_k^2 or e_k —this means that H is connected by the center vertex v . If N_2 is nonempty, append H to the share of an arbitrary agent in N_2 . Else, if some segment e_k^1 is allocated to at least two agents, give H to the agent who has been allocated the point x_k . Otherwise, we know that every agent is allocated exactly one segment in E^1 —give H to the agent who traded her share last in Phase 2 (in particular, Phase 2a).

We claim that this algorithm yields a $(3 + \epsilon)$ -EF allocation. By Proposition 1, such an allocation is roughly $1/2$ -additive-EF and (by taking $\epsilon = 1/n$) 3 -proportional as well.

Theorem 4. *Given an instance of graphical cake cutting consisting of a star graph with m edges, there exists an algorithm that, for any $\epsilon > 0$, computes a $(3 + \epsilon)$ -EF allocation in time polynomial in n, m , and $1/\epsilon$.*

4 Identical Valuations

In this section, we focus on the case where the valuation functions of all agents are identical. While this case is uninteresting for interval cake cutting since a fully envy-free allocation can be trivially found, it becomes highly nontrivial when graphs are involved (see, for example, the left of Figure 1). Indeed, a number of works on dividing edges or vertices of a graph can be interpreted as dealing with the identical-valuation setting [Wu *et al.*, 2007; Chu *et al.*, 2010; Caragiannis *et al.*, 2022]. Moreover, this setting captures scenarios where there is an objective measure across agents, for example, when a town wants to divide the responsibility of maintaining its streets among contractors based on the lengths or numbers of residents on the streets.

As we mentioned in Section 2, an α -EF allocation is not guaranteed to exist for any $\alpha < 2$, even with identical valuations and star graphs. We will show in this section that, for arbitrary graphs and any $\epsilon > 0$, it is possible to find an allocation that is $(2 + \epsilon)$ -EF, which means that the approximation factor of 2 is essentially tight.

To this end, we first discuss how we can find a 4-EF allocation using a variation of the ITERATIVEDIVIDE algorithm that we saw in Section 3. This 4-EF allocation will later be used as an input to an algorithm that computes a $(2 + \epsilon)$ -EF allocation. For star graphs, we also describe a simpler method for computing a 2-EF allocation.

Let us denote by μ the common valuation function of the agents, and define $\max(\mathcal{A}) = \max_{i \in N} \mu(A_i)$ and $\min(\mathcal{A}) = \min_{i \in N} \mu(A_i)$ for any allocation $\mathcal{A} = (A_1, \dots, A_n)$.

4.1 4-EF

In ITERATIVEDIVIDE, we used the threshold $\beta = 1/4$ in every call to DIVIDE so as to allocate a share worth at least $1/4$ to some agent, which results in a $1/2$ -additive-EF allocation. Even with identical valuations, each iteration of DIVIDE is unpredictable in the sense that the recipient could receive a share worth anywhere between β and 2β . If β is chosen to be more than $1/(2n - 2)$ and the first $n - 1$ agents all take shares of value close to 2β , then the last agent will be left effectively empty-handed. In contrast, if β is chosen to be at most $1/(2n - 2)$ and the first $n - 1$ agents all take shares of value only β , then the last agent will receive a share of value at least $1/2$, which leads to an envy factor linear in n .

To resolve this problem, let us consider using an *adaptive* threshold that takes the values of the previous shares into account. If the previous agents took large shares, then the threshold β is reduced appropriately for the current agent, and vice versa. Without loss of generality, assume that for each $i \in \{1, \dots, n - 1\}$, agent i is the one who takes the first share generated by the i^{th} iteration of DIVIDE. By choosing $\beta_i = \frac{1}{2}(\frac{2i}{2n-1} - \sum_{j=1}^{i-1} \mu(A_j))$ to be the threshold for the i^{th} iteration of DIVIDE, we claim that the resulting allocation is 4-EF.⁷ Along the way, we shall see that the allocation is also $(2 - 1/n)$ -proportional.

⁷Even better, the allocation is actually $(4 - 1/2^{n-3})$ -EF, as shown in the proof of Theorem 5 [Yuen and Suksompong, 2023].

Algorithm 2 RECURSIVEBALANCE(\mathcal{A}, ϵ).

Input: Allocation $\mathcal{A} = (A_1, \dots, A_n)$, $\epsilon \in (0, 1)$.

Output: Allocation $\mathcal{A} = (A_1, \dots, A_n)$.

```

1: while  $\frac{\max(\mathcal{A})}{\min(\mathcal{A})} > 2 + \epsilon$  do
2:    $\mathcal{A} \leftarrow \text{BALANCE}(\mathcal{A}, \epsilon)$ 
3: end while
4: return  $\mathcal{A}$ 
    
```

Algorithm 3 BALANCE(\mathcal{A}, ϵ).

Input: Allocation $\mathcal{A} = (A_1, \dots, A_n)$, $\epsilon \in (0, 1)$.

Output: Allocation $\mathcal{A} = (A_1, \dots, A_n)$.

```

1:  $\mathcal{P} \leftarrow \text{MINMAXPATH}(\mathcal{A})$ 
2:  $\mathcal{A} \leftarrow (\mathcal{A} \setminus \mathcal{P}) \cup \text{BALANCEPATH}(\mathcal{P}, \epsilon)$ 
3: return  $\mathcal{A}$ 
    
```

Theorem 5. *Given an instance of graphical cake cutting consisting of n agents with identical valuations, there exists an algorithm that computes a 4-EF and $(2 - \frac{1}{n})$ -proportional allocation in time polynomial in n and the size of G .*

As mentioned in Section 2, an α -proportional allocation is not guaranteed to exist for any $\alpha < 2 - 1/n$, so the algorithm in Theorem 5 attains the optimal proportionality approximation. In terms of envy-freeness, one may consider adjusting the values of β_i in ITERATIVEDIVIDE so as to obtain an allocation with a better approximation factor than 4. While this might be possible, it seems unlikely that this approach could lead to 2-EF, given that the guarantee from Lemma 2 already has a multiplicative gap of 2. This motivates us to devise another algorithm that reduces the factor to arbitrarily close to 2.

4.2 $(2 + \epsilon)$ -EF

Let us first define a *minimum-maximum path* of an allocation $\mathcal{A} = (A_1, \dots, A_n)$ as a list $\mathcal{P} = (P_1, \dots, P_d)$ (where $d \leq n$) satisfying the following conditions:

- For each $i \in \{1, \dots, d\}$, $P_i = A_j$ for some $j \in \{1, \dots, n\}$,
- $P_i \neq P_j$ for $1 \leq i < j \leq d$,
- For each $i \in \{1, \dots, d - 1\}$, there exists at least one point belonging to both P_i and P_{i+1} , and
- $\mu(P_1) = \min(\mathcal{A})$ and $\mu(P_d) = \max(\mathcal{A})$.

Intuitively, a minimum-maximum path is a list of shares that chains a minimum-valued one to a maximum-valued one in the underlying graph without crossing any share more than once. Such a list can be found in polynomial time: we locate a minimum-valued share and a maximum-valued share of \mathcal{A} , find a path through the graph that connects both shares, and identify the shares corresponding to this path. If some share A_i appears more than once, we repeatedly remove the part between the two occurrences of A_i (including one of these occurrences). Let us use $\text{MINMAXPATH}(\mathcal{A})$ to denote an arbitrary minimum-maximum path of \mathcal{A} .

We now describe the algorithm, RECURSIVEBALANCE (Algorithm 2), which finds a $(2 + \epsilon)$ -EF allocation for

Algorithm 4 BALANCEPATH(\mathcal{P}, ϵ).

Input: List of shares $\mathcal{P} = (P_1, \dots, P_d)$, $\epsilon \in (0, 1)$.

Output: List of d shares (P_1, \dots, P_d) .

Initialization: $\gamma \leftarrow \mu(P_d)$.

```

1: for  $i = 1, \dots, d - 1$  do
2:   if  $\mu(P_i) \geq \frac{\gamma}{2 + \epsilon}$  then
3:     return  $(P_1, \dots, P_d)$ 
4:   end if
5:    $P^* \leftarrow P_i \cup P_{i+1}$ 
6:   if  $\mu(P^*) < \frac{2\gamma}{2 + \epsilon}$  then
7:      $P_i \leftarrow P^*$ 
8:      $r \leftarrow$  any vertex in  $P_d$ 
9:      $(P_{i+1}, P_d) \leftarrow \text{DIVIDE}(P_d, N, \frac{\gamma}{3}, r)$ 
10:    return  $(P_1, \dots, P_d)$ 
11:  end if
12:  if  $i = d - 1$  then
13:     $r \leftarrow$  any vertex in  $P_d$ 
14:  else
15:     $r \leftarrow$  any vertex in  $P_{i+1} \cap P_{i+2}$ 
16:  end if
17:   $(P_i, P_{i+1}) \leftarrow \text{DIVIDE}(P^*, N, \frac{\gamma}{2 + \epsilon}, r)$ 
18: end for
19: return  $(P_1, \dots, P_d)$ 
    
```

any given $\epsilon > 0$. We employ similar ideas as the ones used by Chu *et al.* [2010]—in their work, they partition the (indivisible) edges of a graph, whereas we have to account for the divisibility of the edges. Assume without loss of generality that $\epsilon \in (0, 1)$. Given an allocation \mathcal{A} , RECURSIVEBALANCE repeatedly replaces \mathcal{A} with the allocation BALANCE(\mathcal{A}, ϵ), then terminates when \mathcal{A} is $(2 + \epsilon)$ -EF. The algorithm BALANCE (Algorithm 3) finds a minimum-maximum path \mathcal{P} of \mathcal{A} , and replaces the shares in \mathcal{A} that appear in \mathcal{P} by BALANCEPATH(\mathcal{P}, ϵ). Note that the order of shares in \mathcal{A} does not affect the fairness properties due to the identical valuation across all agents.

Now, BALANCEPATH (Algorithm 4) does the bulk of the work. This algorithm adjusts the shares in $\mathcal{P} = (P_1, \dots, P_d)$ so that their values meet certain criteria. Let $\gamma = \mu(P_d)$ and $\hat{P}_1 = P_1$.⁸ For each i from 1 to $d - 1$, the algorithm does one of the following unless it is terminated prematurely via Case 1 or Case 2.

- **Case 1: The value of \hat{P}_i is at least $\frac{\gamma}{2 + \epsilon}$.**

Set $P'_i = \hat{P}_i$ and $P'_j = P_j$ for all $j \in \{i + 1, \dots, d\}$. Terminate the algorithm by returning (P'_1, \dots, P'_d) .

- **Case 2: The value of \hat{P}_i is less than $\frac{\gamma}{2 + \epsilon}$ and the value of $\hat{P}_i \cup P_{i+1}$ is less than $\frac{2\gamma}{2 + \epsilon}$.**

Set $P'_i = \hat{P}_i \cup P_{i+1}$. Set (P'_{i+1}, P'_d) to be the output of $\text{DIVIDE}(P_d, N, \frac{\gamma}{3}, r)$, where r is any point in P_d ; note that this call to DIVIDE is valid because P_d has value γ . Set $P'_j = P_j$ for all $j \in \{i + 2, \dots, d - 1\}$. Terminate

⁸We reuse P_i for P'_i and \hat{P}_i in the pseudocode for simplicity; however, we differentiate them in the main text for clarity.

the algorithm by returning (P'_1, \dots, P'_d) .

- **Case 3: The value of \widehat{P}_i is less than $\frac{\gamma}{2+\epsilon}$ and the value of $\widehat{P}_i \cup P_{i+1}$ is at least $\frac{2\gamma}{2+\epsilon}$.**

Consider the graph $P^* = \widehat{P}_i \cup P_{i+1}$. Set $(P'_i, \widehat{P}_{i+1})$ to be the output of $\text{DIVIDE}(P^*, N, \frac{\gamma}{2+\epsilon}, r)$, where r is any point belonging to both P_{i+1} and P_{i+2} (unless $i = d-1$, in which case r is any point in P_d); note that this call to DIVIDE is valid because P^* has value at least $\frac{2\gamma}{2+\epsilon}$. The choice of r ensures that, if $i < d-1$, \widehat{P}_{i+1} and P_{i+2} share at least one point. Continue with the next i by incrementing it by 1.

If the algorithm still has not terminated after $i = d-1$, set $P'_d = \widehat{P}_d$ and return (P'_1, \dots, P'_d) .

We claim that the algorithm RECURSIVEBALANCE terminates in polynomial time if it receives a 4-EF allocation as input (provided by Theorem 5), and upon termination the algorithm returns a $(2 + \epsilon)$ -EF allocation.

Theorem 6. *Given an instance of graphical cake cutting consisting of n agents with identical valuations, there exists an algorithm that, for any $\epsilon > 0$, computes a $(2 + \epsilon)$ -EF allocation in time polynomial in n , $1/\epsilon$, and the size of G .*

4.3 Star Graphs

Although RECURSIVEBALANCE provides a nearly 2-EF allocation, the algorithm is rather complex and involves many steps. Moreover, it seems difficult to improve the guarantee to *exactly* 2-EF via the algorithm, as the balancing of shares may be too slow and we may not reach a 2-EF allocation in finite time.⁹ For the case of star graphs, it is possible to obtain a 2-EF allocation using a simpler “bag-filling” approach.

Theorem 7. *Given an instance of graphical cake cutting consisting of a star graph with m edges and n agents with identical valuations, there exists an algorithm that computes a 2-EF allocation in time polynomial in n and m .*

5 Beyond One Connected Piece

In previous sections, we made the crucial assumption that each agent necessarily receives a connected piece of the graphical cake. We now relax this assumption and explore improvements in envy-freeness guarantees if each agent is allowed to receive a small number of connected pieces. As mentioned in Section 1.2, extensive research has been done on (approximate) envy-free connected allocations of an *interval cake*; it will be interesting to draw connections between these results and graphical cake cutting.

As an example, consider a star graph with edges e_k for $k \in \{1, \dots, m\}$. Rearrange the edges to form a path graph with the edges e_1 to e_m going from left to right; the farthest end of each edge from the center vertex in the star graph is oriented towards the *right* on the path graph. Note that any segment along the path graph corresponds to at most

⁹For example, agents with shares of 1 unit and $2 + \epsilon$ units respectively may repeatedly exchange and adjust their shares with each other (through multiple calls to Algorithm 4) but may never reach a multiplicative envy factor of 2.

two connected pieces in the star graph.¹⁰ It follows that if a connected α -EF (resp., α -additive-EF) allocation of an interval cake can be found in polynomial time, then there also exists a polynomial-time algorithm that finds an α -EF (resp., α -additive-EF) allocation of the star cake where each agent receives at most *two* connected pieces.

We now consider an arbitrary graph G . Let ϕ be a bijection of the edges of G onto the edges of a path graph with the same number of edges as G , where the orientation of the edges along the path can be chosen. By an abuse of notation, let $\phi(G)$ be the corresponding path graph. Define the *path similarity number of G associated with ϕ* , denoted by $\text{PSN}(G, \phi)$, as the smallest number k such that any segment along $\phi(G)$ corresponds to at most k connected pieces in G . Any results pertaining to connected allocations of an interval cake¹¹ can be directly applied to allocations of G in which each agent receives at most $\text{PSN}(G, \phi)$ connected pieces.

Proposition 8. *Let G be a graph and ϕ be a bijection of the edges of G onto $\phi(G)$ with orientation such that ϕ can be found in polynomial time. If there exists a polynomial-time algorithm that computes a connected α -EF allocation of an interval cake, then there exists a polynomial-time algorithm that computes an α -EF allocation of a graphical cake G where each agent receives at most $\text{PSN}(G, \phi)$ connected pieces. An analogous statement holds for α -additive-EF.*

To complement Proposition 8, we provide upper bounds of $\text{PSN}(G, \phi)$ where ϕ can be computed in polynomial time.

Theorem 9. *Let G be a tree of height h . Then there exists a bijection ϕ that can be computed in polynomial time such that $\text{PSN}(G, \phi) \leq h + 1$.*

Theorem 10. *Let G be a connected graph, and let d be the diameter of a spanning tree of G with the minimum diameter. Then there exists a bijection ϕ that can be computed in polynomial time such that $\text{PSN}(G, \phi) \leq \lceil d/2 \rceil + 2$.*

6 Conclusion and Future Work

In this paper, we have studied the existence and computation of approximately envy-free allocations in graphical cake cutting. An interesting question left open by our work is whether a connected allocation with a constant multiplicative approximation of envy-freeness can be guaranteed for general graphs and non-identical valuations—the techniques that we developed for star graphs (Section 3.2) do not seem sufficient for answering this question. If a non-connected allocation is allowed, then tightening the bounds for the path similarity number (Section 5) will reduce the number of connected pieces each agent receives to achieve the same envy-freeness approximation. Another intriguing direction for non-connected allocations is whether we can obtain improved envy bounds using a different approach than converting the graph into a path.

¹⁰A segment can start and end at points within an edge and span across different edges—it need not start and end at vertices.

¹¹For instance, Barman and Kulkarni [2022] presented a polynomial-time algorithm that computes a (roughly) $1/4$ -additive-EF and 2-EF allocation of an interval cake.

Acknowledgments

This work was partially supported by the Singapore Ministry of Education under grant number MOE-T2EP20221-0001 and by an NUS Start-up Grant. We thank the anonymous reviewers for their valuable feedback.

References

- [Arunachaleswaran *et al.*, 2019] Eshwar Ram Arunachaleswaran, Siddharth Barman, Rachitesh Kumar, and Nidhi Rathi. Fair and efficient cake division with connected pieces. In *Proceedings of the 15th Conference on Web and Internet Economics (WINE)*, pages 57–70, 2019. Extended version available at CoRR abs/1907.11019v4.
- [Aziz and Mackenzie, 2016] Haris Aziz and Simon Mackenzie. A discrete and bounded envy-free cake cutting protocol for any number of agents. In *Proceedings of the 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 416–427, 2016.
- [Barman and Kulkarni, 2022] Siddharth Barman and Pooja Kulkarni. Approximation algorithms for envy-free cake division with connected pieces. *CoRR*, abs/2208.08670, 2022.
- [Bei and Suksompong, 2021] Xiaohui Bei and Warut Suksompong. Dividing a graphical cake. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5159–5166, 2021.
- [Bei *et al.*, 2012] Xiaohui Bei, Ning Chen, Xia Hua, Biaoshuai Tao, and Endong Yang. Optimal proportional cake cutting with connected pieces. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1263–1269, 2012.
- [Brams and Taylor, 1996] Steven J. Brams and Alan D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.
- [Caragiannis *et al.*, 2022] Ioannis Caragiannis, Evi Micha, and Nisarg Shah. A little charity guarantees fair connected graph partitioning. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, pages 4908–4916, 2022.
- [Cechlárová *et al.*, 2013] Katarína Cechlárová, Jozef Doboš, and Eva Pillárová. On the existence of equitable cake divisions. *Information Sciences*, 228:239–245, 2013.
- [Chu *et al.*, 2010] An-Chiang Chu, Bang Ye Wu, Hung-Lung Wang, and Kun-Mao Chao. A tight bound on the min-ratio edge-partitioning problem of a tree. *Discrete Applied Mathematics*, 158(14):1471–1478, 2010.
- [Deligkas *et al.*, 2022] Argyrios Deligkas, Eduard Eiben, Robert Ganian, Thekla Hamm, and Sebastian Ordyniak. The complexity of envy-free graph cutting. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 237–243, 2022.
- [Dubins and Spanier, 1961] Lester E. Dubins and Edwin H. Spanier. How to cut a cake fairly. *American Mathematical Monthly*, 68(1):1–17, 1961.
- [Elkind *et al.*, 2021] Edith Elkind, Erel Segal-Halevi, and Warut Suksompong. Graphical cake cutting via maximin share. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 161–167, 2021.
- [Goldberg *et al.*, 2020] Paul W. Goldberg, Alexandros Hollender, and Warut Suksompong. Contiguous cake cutting: Hardness results and approximation algorithms. *Journal of Artificial Intelligence Research*, 69:109–141, 2020.
- [Igarashi and Zwicker, 2021] Ayumi Igarashi and William S. Zwicker. Fair division of graphs and of tangled cakes. *CoRR*, abs/2102.08560, 2021.
- [Procaccia, 2013] Ariel D. Procaccia. Cake cutting: Not just child’s play. *Communications of the ACM*, 56(7):78–87, 2013.
- [Procaccia, 2016] Ariel D. Procaccia. Cake cutting algorithms. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 13, pages 311–329. Cambridge University Press, 2016.
- [Robertson and Webb, 1998] Jack Robertson and William Webb. *Cake-Cutting Algorithms: Be Fair if You Can*. Peters/CRC Press, 1998.
- [Stromquist, 1980] Walter Stromquist. How to cut a cake fairly. *American Mathematical Monthly*, 87(8):640–644, 1980.
- [Stromquist, 2008] Walter Stromquist. Envy-free cake divisions cannot be found by finite protocols. *The Electronic Journal of Combinatorics*, 15:#R11, 2008.
- [Su, 1999] Francis Edward Su. Rental harmony: Sperner’s lemma in fair division. *American Mathematical Monthly*, 106(10):930–942, 1999.
- [Suksompong, 2021] Warut Suksompong. Constraints in fair division. *ACM SIGecom Exchanges*, 19(2):46–61, 2021.
- [Wu *et al.*, 2007] Bang Ye Wu, Hung-Lung Wang, Shih Ta Kuan, and Kun-Mao Chao. On the uniform edge-partition of a tree. *Discrete Applied Mathematics*, 155(10):1213–1223, 2007.
- [Yuen and Suksompong, 2023] Sheung Man Yuen and Warut Suksompong. Approximate envy-freeness in graphical cake cutting. *CoRR*, abs/2304.11659, 2023.