

Almost Envy-Free Allocations with Connected Bundles

Vittorio Bilò

University of Salento, Lecce, Italy

Ioannis Caragiannis

University of Patras, Rion-Patras, Greece

Michele Flammini

Gran Sasso Science Institute and University of L'Aquila, L'Aquila, Italy

Ayumi Igarashi

Kyushu University, Fukuoka, Japan

Gianpiero Monaco

University of L'Aquila, L'Aquila, Italy

Dominik Peters

University of Oxford, Oxford, U.K.

Cosimo Vinci

University of L'Aquila, L'Aquila, Italy

William S. Zwicker

Union College, Schenectady, USA

Abstract

We study the existence of allocations of indivisible goods that are envy-free up to one good (EF1), under the additional constraint that each bundle needs to be connected in an underlying item graph G . When the items are arranged in a path, we show that EF1 allocations are guaranteed to exist for arbitrary monotonic utility functions over bundles, provided that either there are at most four agents, or there are any number of agents but they all have identical utility functions. Our existence proofs are based on classical arguments from the divisible cake-cutting setting, and involve discrete analogues of cut-and-choose, of Stromquist's moving-knife protocol, and of the Su-Simmons argument based on Sperner's lemma. Sperner's lemma can also be used to show that on a path, an EF2 allocation exists for any number of agents. Except for the results using Sperner's lemma, all of our procedures can be implemented by efficient algorithms. Our positive results for paths imply the existence of connected EF1 or EF2 allocations whenever G is traceable, i.e., contains a Hamiltonian path. For the case of two agents, we completely characterize the class of graphs G that guarantee the existence of EF1 allocations as the class of graphs whose biconnected components are arranged in a path. This class is strictly larger than the class of traceable graphs; one can check in linear time whether a graph belongs to this class, and if so return an EF1 allocation.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory, Mathematics of computing → Combinatoric problems, Mathematics of computing → Graph theory

Keywords and phrases Envy-free Division, Cake-cutting, Resource Allocation, Algorithmic Game Theory

Digital Object Identifier 10.4230/LIPIcs.ITCS.2019.14



© Vittorio Bilò, Ioannis Caragiannis, Michele Flammini, Ayumi Igarashi, Gianpiero Monaco, Dominik Peters, Cosimo Vinci, and William S. Zwicker;
licensed under Creative Commons License CC-BY

10th Innovations in Theoretical Computer Science (ITCS 2019).

Editor: Avrim Blum; Article No. 14; pp. 14:1–14:21



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Related Version [5], <https://arxiv.org/abs/1808.09406> with omitted proofs, and additional results

Funding D. Peters is supported by ERC grant 639945 (ACCORD). A. Igarashi is supported by KAKENHI (Grant-in-Aid for JSPS Fellows, No. 18J00997) Japan. While working on this paper, W. S. Zwicker was supported by the Oliver Smithies Visiting Fellowship at Balliol College, Oxford.

1 Introduction

A famous literature considers the problem of *cake-cutting* [10, 25, 24]. There, a divisible heterogeneous resource (a *cake*, usually formalized as the interval $[0, 1]$) needs to be divided among n agents. Each agent has a valuation function over subsets of the cake, usually formalized as an atomless measure over $[0, 1]$. The aim is to partition the cake into n pieces, and allocate each piece to one agent, in a “fair” way. By fair, we will mean that the allocation is *envy-free*: no agent thinks that another agent’s piece is more valuable than her own.

When there are two agents, the classic procedure of cut-and-choose can produce an envy-free division: a knife is moved from left to right, until an agent shouts to indicate that she thinks the pieces to either side are equally valuable. The other agent then picks one of the pieces, leaving the remainder for the shouter. As is easy to see, the result is an envy-free allocation. For three or more agents, finding an envy-free division has turned out to be much trickier. An early result by Dubins and Spanier [15] used Lyapunov’s Theorem and measure-theoretic techniques to show, non-constructively, that an envy-free allocation always exists. However, as Stromquist [28] memorably writes, “their result depends on a liberal definition of a “piece” of cake, in which the possible pieces form an entire σ -algebra of subsets. A player who only hopes for a modest interval of cake may be presented instead with a countable union of crumbs.” In many applications of resource allocation (such as land division, or the allocation of time slots), agents have little use for a severely disconnected piece of cake.

Stromquist [28] himself offered a solution, and gave a new non-constructive argument (using topology) which proved that there always exists an envy-free division of the cake into *intervals*. Forest Simmons later observed that the proof could be simplified by using Sperner’s lemma, and this technique was subsequently presented in a paper by Su [29]. For the three-agent case, Stromquist [28] also presented an appealing moving-knife procedure that more directly yields a connected envy-free allocation. For $n \geq 4$ agents, no explicit procedures are known to produce a connected envy-free allocation (i.e., an allocation where the cake is cut in exactly $n - 1$ places). However, for $n = 4$, several moving-knife procedures exist that only need few cuts; for example, the Brams–Taylor–Zwicker [11] procedure requires 11 cuts, and a protocol of Barnabel and Brams [3] requires 5 cuts.

In many applications, the resources to be allocated are not infinitely divisible, and we face the problem of allocating *indivisible goods*. Most of the literature on indivisible goods has not assumed any kind of structure on the item space, in contrast to the rich structure of the interval $[0, 1]$ in cake-cutting. Thus, there has been little attention on minimizing the number of “cuts” required in an allocation. However, when the items have a spatial or temporal structure, this consideration is important.

In this paper, we study the allocation of items that are arranged on a *path* or other structure, and impose the requirement that only *connected* subsets of items may be allocated to the agents. Formally, we work in the model of [9], who assume that the items form the vertex set of a graph G ; a bundle is connected if it induces a connected subgraph of G . In

their paper, it became apparent that techniques from cake-cutting can be usefully ported to achieve good allocations in the indivisible case. For example, moving-knife procedures that achieve proportionality in cake-cutting have analogues that produce allocations that satisfy the *maximin share guarantee* [12].¹

Do envy-free procedures for cake-cutting also translate to the indivisible case? Of course, in general, it is impossible to achieve envy-freeness with indivisibilities (consider two agents and a single desirable item), but we can look for approximations. A relaxation of envy-freeness that has been very influential recently is envy-freeness *up to one good* (EF1), introduced by Budish [12]. It requires that an agent's envy towards another bundle vanishes if we remove some item from the envied bundle. In the setting without connectivity constraints and with additive valuations, the maximum Nash welfare solution satisfies EF1, as does a simple round-robin procedure [13]. The well-known envy-graph algorithm [21] also guarantees EF1. However, none of these procedures respects connectivity constraints.

When items are arranged on a path, we prove that connected EF1 allocations exist when there are two, three, or four agents. As was necessary in cake-cutting, we use successively more complicated tools to establish these existence results. For two agents, there is a discrete analogue of cut-and-choose that satisfies EF1. In that procedure, a knife moves across the path, and an agent shouts when the knife reaches what we call a *lumpy tie*, that is when the bundles to either side of the knife have equal value *up to one item*. For three agents, we design an algorithm mirroring Stromquist's moving-knife procedure which guarantees EF1. For four agents, we show that Sperner's lemma can be used to prove that an EF1 allocation exists, via a technique inspired by the Simmons–Su approach, and an appropriately triangulated simplex of connected partitions of the path. For five or more agents, we were not able to establish the existence of EF1 allocations on a path, but we can show (again via Sperner's lemma) that EF2 allocations exist, strengthening a prior result of Suksompong [30]. We also show that if all agents have the same valuation function over bundles, then an egalitarian-welfare-optimal allocation, after suitably reallocating some items, is EF1.

These existence results require only that agents' valuations are monotonic (they need not be additive), and in addition ensure that the constructed allocation satisfies the maximin share guarantee (see [5]). Moreover, the fairness guarantee of our algorithms is slightly stronger than the standard notion of EF1: in the returned allocations, envy can be avoided by removing just an *outer* item – one whose removal leaves the envied bundle connected. Computationally speaking, all our existence results are immediately useful, since an example of an EF1 allocation can be found by iterating through all $O(m^n)$ connected allocation (this stands in contrast to cake-cutting where we cannot iterate through all possibilities). While we know of no faster algorithms to obtain an EF1 allocation in the cases where we appeal to Sperner's lemma, our other procedures can all be implemented efficiently.

In simultaneous and independent work, Oh et al. [23] designed protocols to find EF1 allocations in the setting without connectivity constraints, aiming for low *query complexity*. They found that adapting cake-cutting protocols to the setting of indivisible items arranged on a path is an especially potent way to achieve low query complexity. This led them to also study a discrete version of the cut-and-choose protocol which achieves connected EF1 allocations for two agents, and they found an alternative proof that an EF1 allocation on a path always exists with identical valuations. They also present a discrete analogue of the Selfridge–Conway procedure which, for three agents with additive valuations, produces an

¹ Another paper by Suksompong [30] works in the same model, and also found that procedures for proportionality and other concepts can be applied to the indivisible setting.

allocation of a path into bundles that have a constant number of connected components. However, they do not study connected allocations on graphs that are not paths, and they do not consider the case of (non-identical) general valuations with more than two agents.

A recurring theme in our algorithms is the specific way that the moving knives from cake-cutting are rendered in the discrete setting. While one might expect knives to be placed over the edges of the path, and “move” from edge to edge, we find that this movement is too “fast” to ensure EF1 (see also footnote 4). Instead, our knives alternate between hovering over edges and items. When a knife hovers over an item, we imagine the knife’s blade to be “thick”: the knife *covers* the item, and agents then pretend that the covered item does not exist. These intermediate steps are useful, since they can tell us that envy will vanish if we hide an item from a bundle.

What about graphs G other than paths? Our positive results for paths immediately generalize to traceable graphs (those that contain a Hamiltonian path), since we can run the algorithms pretending that the graph only consists of the Hamiltonian path. For the two-agent case, we completely characterize the class of graphs that guarantee the existence of EF1 allocations: Our discrete cut-and-choose protocol can be shown to work on all graphs G that admit a *bipolar numbering*, which exists if and only if the biconnected components (blocks) of G can be arranged in a path. By constructing counterexamples, we prove that no graph failing this condition (for example, a star) guarantees EF1, even for identical, additive, binary valuations. For the case of three or more agents, it is a challenging open problem to characterize the class of graphs guaranteeing EF1 (or even to find an infinite class of non-traceable graphs that guarantees EF1).

2 Preliminaries

For each natural number $s \in \mathbb{N}$, write $[s] = \{1, 2, \dots, s\}$.

Let $N = [n]$ be a finite set of *agents* and $G = (V, E)$ be an undirected finite graph. The vertices in V as *goods* or *items*. A subset I of V is *connected* if it induces a connected subgraph of G . We write $\mathcal{C}(V)$ for the set of connected subsets of V . We call a set $I \in \mathcal{C}(V)$ a (connected) *bundle*. Each agent $i \in N$ has a *valuation function* $u_i : \mathcal{C}(V) \rightarrow \mathbb{R}$ over connected bundles, which we will always assume to be *monotonic*, that is, $X \subseteq Y$ implies $u_i(X) \leq u_i(Y)$. We also assume that $u_i(\emptyset) = 0$ for each $i \in N$. Monotonicity implies that items are *goods*; we do not consider *bads* (or *chores*) in this paper. We say that an agent $i \in N$ *weakly prefers* bundle X to bundle Y if $u_i(X) \geq u_i(Y)$.² A (connected) *allocation* $A : N \rightarrow \mathcal{C}(V)$ assigns each agent $i \in N$ a connected bundle $A(i) \in \mathcal{C}(V)$ such that each item occurs in exactly one bundle, i.e., $\bigcup_{i \in N} A(i) = V$ and $A(i) \cap A(j) = \emptyset$ when $i \neq j$.

We say that the agents have *identical valuations* if for all $i, j \in N$ and every bundle $I \in \mathcal{C}(V)$, we have $u_i(I) = u_j(I)$. A valuation function u_i is *additive* if $u_i(I) = \sum_{v \in I} u_i(\{v\})$ for each bundle $I \in \mathcal{C}(V)$. Many examples in this paper will use identical additive valuations, and will take G to be a path. In this case, we use a shorthand to specify these examples; the meaning of this notation should be clear. For example, we write “2–1–3–1” to denote an instance with four items v_1, v_2, v_3, v_4 arranged on a path, and where $u_i(\{v_1\}) = 2, \dots, u_i(\{v_4\}) = 1$ for each i . For such an instance, an allocation will be written as a tuple, e.g., (2, 1–3–1) denoting an allocation allocating bundles $\{v_1\}$ and $\{v_2, v_3, v_4\}$, noting that with identical valuations it does not usually matter which agent receives which bundle.

² Our arguments only operate based on agents’ ordinal preferences over bundles, and the (cardinal) valuation functions are only used for notational convenience. One exception, perhaps, is in Algorithm 1 where we calculate a leximin allocation, but the algorithm can be applied after choosing an arbitrary utility function consistent with the ordinal preferences.

An allocation A is *envy-free* if $u_i(A(i)) \geq u_i(A(j))$ for every pair $i, j \in N$ of agents, that is, if every agent thinks that their bundle is a best bundle in the allocation. It is well-known that an envy-free allocation may not exist (consider two agents and one good). The main fairness notion that we study is a version of *envy-freeness up to one good* (EF1), a relaxation of envy-freeness introduced by Budish [12], adapted to the model with connectivity constraints. This property states that an agent i will not envy another agent j after we remove some item from j 's bundle. Since we only allow connected bundles in our set-up, we may only remove an item from $A(j)$ if removal of this item leaves the bundle connected.

► **Definition 1** (EF1: envy-freeness up to one *outer* good). An allocation A satisfies *EF1* if for any pair $i, j \in N$ of agents, either $A(j) = \emptyset$ or there is a good $v \in A(j)$ such that $A(j) \setminus \{v\}$ is connected and $u_i(A(i)) \geq u_i(A(j) \setminus \{v\})$.

In the instance 2–1–3–1 for two agents, the allocation (2–1, 3–1) is EF1, since the left agent's envy can be eliminated by removing the item of value 3 from the right-hand bundle. However, the allocation (2, 1–3–1) fails to be EF1 according to our definition, since eliminating either outer good of the right bundle does not prevent envy.³

► **Definition 2.** A graph G *guarantees EF1* (for a specific n) if for all possible monotonic valuations for n agents, there exists some connected allocation that is EF1.

For reasoning about EF1 allocations, let us introduce a few shorthands. Given an allocation A we will say that $i \in N$ *does not envy* $j \in N$ *up to* v if $u_i(A(i)) \geq u_i(A(j) \setminus \{v\})$. The *up-to-one valuation* $u_i^- : \mathcal{C}(V) \rightarrow \mathbb{R}_{\geq 0}$ of agent $i \in N$ is defined, for every $I \in \mathcal{C}(V)$, as

$$u_i^-(I) := \begin{cases} 0 & \text{if } I = \emptyset, \\ \min \{u_i(I \setminus \{v\}) : v \in I \text{ such that } I \setminus \{v\} \text{ is connected}\} & \text{if } I \neq \emptyset. \end{cases} \quad (1)$$

Thus, an allocation A satisfies EF1 iff $u_i(A(i)) \geq u_i^-(A(j))$ for any pair $i, j \in N$ of agents.

Given an ordered sequence of the vertices $P = (v_1, v_2, \dots, v_m)$, and $j, k \in [m]$ with $j \leq k$, we write $P(v_j, v_k)$ for the subsequence from v_j to v_k , so $P(v_j, v_k) = (v_j, v_{j+1}, \dots, v_{k-1}, v_k)$. Let $L(v_j) = P(v_1, v_{j-1})$ be the subsequence of vertices strictly left of v_j and $R(v_j) = P(v_{j+1}, v_m)$ be the subsequence of vertices strictly right of v_j . A *Hamiltonian path* of a graph G is a path that visits all the vertices of the graph exactly once. A graph is *traceable* if it contains a Hamiltonian path.

3 EF1 existence for two agents

In cake-cutting for two agents, the standard way of obtaining an envy-free allocation is the cut-and-choose protocol: Alice divides the cake into two equally-valued pieces, and Bob selects the piece he prefers; the other piece goes to Alice. The same strategy almost works in the indivisible case when items form a path; the problem is that Alice might not be able to divide the items into two exactly-equal pieces. Instead, we ask Alice to divide the items into pieces that are equally valued “up to one good”. The formal version is as follows. For a sequence of vertices $P = (v_1, v_2, \dots, v_m)$ and an agent i , we say that v_j is the *lumpy tie* over P for agent i if j is the smallest index such that

$$u_i(L(v_j) \cup \{v_j\}) \geq u_i(R(v_j)) \quad \text{and} \quad u_i(R(v_j) \cup \{v_j\}) \geq u_i(L(v_j)). \quad (2)$$

³ This example shows that our definition is strictly stronger than the standard definition of EF1 without connectivity constraints. In the instance 2–1–3–1, considered without connectivity constraints, the allocation (2, 1–3–1) does satisfy EF1 since in the standard setting we are allowed to remove the middle item (with value 3) of the right bundle.

For example, when i has additive valuations 1–3–2–1–3–1, then the third item (of value 2) is the lumpy tie for i , since $1 + 3 + 2 \geq 1 + 3 + 1$ and $2 + 1 + 3 + 1 \geq 1 + 3$. The lumpy tie always exists: taking j to be the smallest index such that $u_i(L(v_j) \cup \{v_j\}) \geq u_i(R(v_j))$ (which exists as the inequality holds for $j = m$ by monotonicity), the first part of (2) holds. If $j = 1$, the second part of (2) is immediate by monotonicity. If $j > 1$, then since j is minimal, we have $u_i(L(v_j)) = u_i(L(v_{j-1}) \cup \{v_{j-1}\}) < u_i(R(v_{j-1})) = u_i(R(v_j) \cup \{v_j\})$ as required.

Using lumpy ties, our discrete version of the cut-and-choose protocol is specified as follows.

Discrete cut-and-choose protocol for $n = 2$ agents on a sequence

$P = (v_1, v_2, \dots, v_m)$:

Step 1. Alice selects her lumpy tie v_j over (v_1, v_2, \dots, v_m) .

Step 2. Bob chooses a weakly preferred bundle among $L(v_j)$ and $R(v_j)$.

Step 3. Alice receives the bundle of all the remaining vertices, including v_j .

Intuitively, the protocol allows Alice to select an item v_j that she will receive for sure, with the advice that the two pieces to either side of v_j should have almost equal value to her. Then, Bob is allowed to choose which side of v_j he wishes to receive. In our example with valuations 1–3–2–1–3–1, Alice selects the lumpy tie of value 2, then Bob chooses the bundle 1–3–1 to the right and receives it, and Alice receives the bundle 1–3–2. The result is EF1. This is true in general, and also if valuations are not identical.

► **Proposition 3.** *When G is a path and there are $n = 2$ agents, the discrete cut-and-choose protocol yields an EF1 allocation.*

Proof. Clearly, the protocol returns a connected allocation. The returned allocation satisfies EF1: Bob does not envy Alice up to item v_j , since Bob receives his preferred bundle among $L(v_j)$ and $R(v_j)$. Also, by (2), Alice does not envy Bob, since Alice either receives the bundle $L(v_j) \cup \{v_j\}$ which she weakly prefers to Bob's bundle $R(v_j)$, or she receives the bundle $R(v_j) \cup \{v_j\}$, which she weakly prefers to Bob's bundle $L(v_j)$. ◀

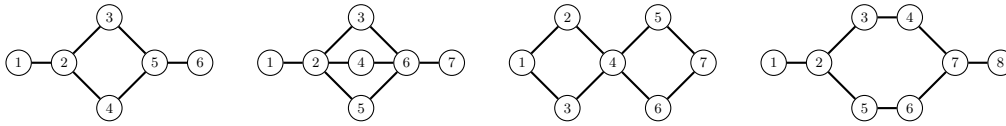
Proposition 3 implies that an EF1 allocation always exists on a path. Hence, an EF1 allocation exists for every traceable graph G : simply use the discrete cut-and-choose protocol on a Hamiltonian path of G . In fact, the discrete cut-and-choose protocol works on a broader class of graphs: We only need to require that the vertices of the graph can be numbered in a way that the allocation resulting from the discrete cut-and-choose protocol is guaranteed to be connected. Since the protocol always partitions the items into an initial and a terminal segment of the sequence, such a numbering needs to satisfy the following property.

► **Definition 4.** A *bipolar numbering* of a graph G is an ordering (v_1, v_2, \dots, v_m) of its vertices such that for all $j \in [n]$, the sets $L(v_j) \cup \{v_j\}$ and $R(v_j) \cup \{v_j\}$ are connected in G .

An equivalent definition (which is the standard one) says that a numbering is bipolar if for every $j \in [n]$, the vertex v_j has a neighbor that appears earlier in the sequence, and a neighbor that appears later in the sequence. Bipolar numberings are used in algorithms for testing planarity and for graph drawing. A Hamiltonian path induces a bipolar numbering, but there are non-traceable graphs that admit a bipolar numbering, see Figure 1 for examples.

► **Proposition 5.** *When there are $n = 2$ agents, then the discrete cut-and-choose protocol run on a bipolar numbering of G yields an EF1 allocation.*

Proof. The discrete cut-and-choose protocol returns an allocation whose bundles are either initial or terminal segments of the ordered sequence (v_1, v_2, \dots, v_m) . By definition of a bipolar numbering, such an allocation is connected, and it is EF1 by Proposition 3. ◀

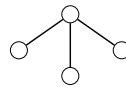


■ **Figure 1** Non-traceable graphs with bipolar numberings.

It is clear that the discrete cut-and-choose protocol cannot be extended to graphs other than those admitting a bipolar numbering. However, it could be that a different protocol is able to produce EF1 allocations on other graphs. In the remainder of this section, we prove that this is not the case: for $n = 2$ agents, a connected graph G guarantees the existence of an EF1 allocation if and only if it admits a bipolar numbering. This completely characterizes the class of graphs that guarantee EF1 existence in the two-agent case.

3.1 Characterization of graphs guaranteeing EF1 for two agents

Based on a known characterization of graphs admitting a bipolar numbering, we characterize this class in terms of forbidden substructures. We then show that these forbidden structures are also forbidden for EF1: if a graph contains such a structure, we can exhibit an additive valuation profile for which no EF1 allocation exists.



As a simple example, consider the star with three leaves, which is the smallest connected graph that does not have a bipolar numbering. Take two agents with identical additive valuations that value each item at 1. Any connected allocation must allocate three items to one agent, and a single item to the other agent. Then the latter agent envies the former agent, even up to one good. This star is an example of what we call a *trident*, and forms a forbidden substructure. Informally, the forbidden substructures take one of two forms. We will prove that a graph G fails to admit a bipolar numbering, and fails to guarantee EF1 for two agents, iff either

- (a) there is a vertex s whose removal from G leaves three or more connected components, or
- (b) there are subgraphs C, P_1, P_2, P_3 of G such that (i) G is the union of these subgraphs, (ii) the subgraphs P_1, P_2, P_3 are vertex-disjoint, (iii) C has exactly one vertex in common with P_i , $i = 1, 2, 3$, and (iv) no edge joins a vertex from one subgraph to a different one.

To reason about these structures, it is useful to consider the *block decomposition* of a graph, which will show that graphs that admit a bipolar numbering have a path-like structure.

► **Definition 6.** A vertex is called a *cut* vertex of a graph G if removing it increases the number of connected components of G . A graph G is *biconnected* if G does not have a cut vertex. A *block* of G is a maximal biconnected subgraph of G .

Equivalently, a block of a graph G can be defined as a maximal subgraph of G where each pair of vertices lie on a common cycle [8]. Given a connected graph G , we define a bipartite graph $B(G)$ with bipartition $(\mathcal{B}, \mathcal{S})$, where \mathcal{B} is the set of blocks of G and \mathcal{S} is the set of cut vertices of a graph G ; a block B and a cut vertex v are adjacent in $B(G)$ if and only if B includes v . Since every cycle of a graph is included in some block, the graph $B(G)$ is a tree:

- **Lemma 7** (e.g., [8]). *Let G be a connected graph. Then*
- *any two blocks of G have at most one cut vertex in common;*
 - *the set of blocks forms a decomposition of G ; and*
 - *the graph $B(G)$ is a tree.*

Thus, for a connected graph G , we call $B(G)$ the *block tree* of G . It turns out that G admits a bipolar numbering if and only if $B(G)$ is a path. For example, the graphs shown in Figure 1 all have their blocks arranged in a path (so that $B(G)$ is a path).

- **Lemma 8.** *A graph G admits a bipolar numbering if its block tree $B(G)$ is a path.*

Proof. Lempel et al. [20] show that G admits a bipolar numbering if there are $s, t \in V$ such that adding an edge $\{s, t\}$ to G makes it biconnected. If $B(G)$ is a path, let B_1 and B_2 be the leaf blocks at the ends of the path $B(G)$. Take any $s \in B_1$ and $t \in B_2$. If we add the edge $\{s, t\}$ to G , the graph becomes biconnected. Hence, G admits a bipolar numbering. ◀

There is a linear-time algorithm based on depth-first search to construct a bipolar numbering for any biconnected graph [16, 31], and one can also calculate the block tree $B(G)$ of a given graph in linear time [17]. Thus, in linear time, we can compute a bipolar numbering of a graph or report that none exists. Clearly, given a bipolar numbering, the discrete cut-and-choose protocol can also be run in linear time.

Next, we show that if $B(G)$ is not a path, then G cannot guarantee EF1. The proof constructs explicit counter-examples, which have a very simple structure. We say that additive valuations u_i are *binary* if $u_i(\{v\}) \in \{0, 1\}$ for every $v \in V$.

- **Lemma 9.** *Let G be a connected graph whose block tree $B(G)$ is not a path. Then there exist identical, additive, binary valuations over G for two agents such that no connected allocation is EF1.*

Proof. If $B(G)$ is not a path, then $B(G)$ has a *trident*, i.e., a vertex with at least three neighbors, and thus either

- (a) there is a cut vertex s adjacent to three blocks B_1 , B_2 , and B_3 ; or
- (b) there is a block B adjacent to three different cut vertices s_1 , s_2 , and s_3 .

In either case, we construct identical additive valuations that do not admit an EF1 allocation.

In case (a), for each $i = 1, 2, 3$, choose a vertex v_i from $B_i \setminus \{s\}$. Note that we can choose such $v_i \neq s$ due to the maximality of B_i . The two agents have utility 1 for s , v_1 , v_2 , and v_3 , and 0 for the remaining vertices. Now take any connected allocation (I_1, I_2) . One of the bundles, say I_1 , includes the cut vertex s . Then I_2 can contain at most one of the vertices v_1 , v_2 , v_3 , since I_2 is connected and does not contain s yet any path between distinct v_i and v_j goes through s . Hence $u_i(I_2) \leq 1$. Now, the bundle I_1 contains s and at least two of v_1 , v_2 , v_3 , so $u_i(I_1) \geq 3$. Thus, the allocation is not EF1.

Case (b) is handled similarly; see [5]. ◀

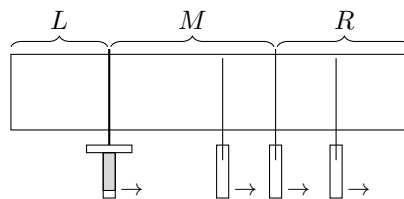
Combining these results, we obtain the promised characterization.

- **Theorem 10.** *The following conditions are equivalent for every connected graph G :*

1. G admits a bipolar numbering.
2. G guarantees EF1 for two agents.
3. G guarantees EF1 for two agents with identical, additive, binary valuations.
4. The block tree $B(G)$ is a path.

The equivalence (2) \Leftrightarrow (3) is noteworthy and perhaps surprising: It is often easier to guarantee fairness when agents' valuations are identical, yet in terms of the graphs that guarantee EF1 for two agents, there is no difference between identical and non-identical valuations. Intriguingly, even for more than two agents, we do not know of a graph which guarantees EF1 for identical valuations, but fails it for non-identical valuations.

4 EF1 existence for three agents: A moving-knife protocol



We will now consider the case of three agents. Stromquist [28] designed a protocol that results in an envy-free contiguous allocation of a divisible cake. In outline, the protocol works as follows. A referee holds a sword over the cake. Each of the three agents holds their own knife over the portion of the cake to the right of the sword, positioning it so that this portion is divided into two pieces they judge to have the same value. Now, initially, the sword is at the left end of the cake. It starts moving at constant speed from left to right, while the agents continuously move their knives to keep dividing the right-hand portion into equally-valued pieces. At some point (when the left-most piece becomes valuable enough), one of the agents shouts “cut”, and the cake will be cut twice: once by the sword, and once by the middle one of the three knives. Agents shout “cut” as soon as the left piece is a highest-valued piece among the three. The agent who shouts receives the left piece. The remaining agents each receive a piece containing their knife. The resulting allocation is envy-free, since the agent receiving the left piece prefers it to the other pieces, and the other agents who are not shouting receive at least half the value of the part of the cake to the right of the sword.

Let G be a path, $P = (v_1, v_2, \dots, v_m)$. There are several difficulties in translating Stromquist’s continuous procedure to the discrete setting for G . First, agents need to divide the piece to the right of the sword in half, and this might not be possible exactly given indivisibilities; but this can be handled using our concept of lumpy ties from Section 3. Next, when the sword moves one item to the right, the lumpy ties of the agents may need to jump several items to the right, for example because the new member of the left-most bundle is very valuable. To ensure EF1, we will need to smoothen these jumps, so that the middle piece grows one item at a time. Also, it will be helpful to have the sword move in half-steps: it alternates between being placed between items (so it cuts the edge between the items), and being placed over an item, in which case the sword covers the item and agents ignore that item. Finally, while the sword covers an item, we will only terminate if at least *two* agents shout to indicate that they prefer the left-most piece; this will ensure that there is an agent who is flexible about which of the bundles they are assigned. The algorithm moves in steps, and alternates between moving the sword, and updating the lumpy ties.

In our formal description of the algorithm, we do not use swords and knives. Instead, we maintain three bundles L , M , and R that can be seen as resulting from a certain configuration of these cutting implements. We also need a few definitions. For a subsequence of vertices $P(v_s, v_r) = (v_s, v_{s+1}, \dots, v_r)$ and an agent i , recall that v_j ($s \leq j \leq r$) is the *lumpy tie* over

$P(v_s, v_r)$ for i if j is the smallest index such that

$$u_i(L(v_j) \cup \{v_j\}) \geq u_i(R(v_j)) \quad \text{and} \quad u_i(R(v_j) \cup \{v_j\}) \geq u_i(L(v_j)). \quad (3)$$

Here, the definitions of $L(v_j)$ and $R(v_j)$ apply to the subsequence $P(v_s, v_r)$. The lumpy tie always exists by the discussion after equation (2). Each of the three agents has a lumpy tie over $P(v_s, v_r)$; a key concept for us is the *median lumpy tie* which is the median of the lumpy ties of the three agents, where the median is taken with respect to the ordering of $P(v_s, v_r)$. We say that $i \in N$ is a *left agent* (respectively, a *middle agent* or a *right agent*) over $P(v_s, v_r)$ if the lumpy tie for i appears strictly before (respectively, is equal to, or appears strictly after) the median lumpy tie. Note that by definition of median, there is at most one left agent, at most one right agent, and at least one middle agent.

Suppose that the median lumpy tie over the subsequence $P(v_s, v_r)$ is v_j , and let i be an agent. Then using the definitions of lumpy tie and left/right agents, we find that

$$\begin{aligned} u_i(L(v_j)) &\geq u_i(R(v_j) \cup \{v_j\}) && \text{if } i \text{ is a left agent, and} \\ u_i(R(v_j)) &\geq u_i(L(v_j) \cup \{v_j\}) && \text{if } i \text{ is a right agent.} \end{aligned} \quad (4)$$

Given the median lumpy tie v_j over $P(v_s, v_r)$, and a two-agent set $S = \{i, k\} \subseteq N$, we define $\text{Lumpy}(S, v_j, P(v_s, v_r))$ to be the allocation of the items in $P(v_s, v_r)$ to S such that

- if i is a left agent and k is a right agent, then i receives $L(v_j)$ and k receives $R(v_j) \cup \{v_j\}$;
- if i is a middle agent, then agent k receives k 's preferred bundle among $L(v_j)$ and $R(v_j)$, and agent i receives the other bundle along with v_j .

Using (3) and (4), we see that $\text{Lumpy}(S, v_j, P(v_s, v_r))$ is an EF1 allocation:

► **Lemma 11** (Median Lumpy Ties Lemma). *Let $S = \{i, k\} \subseteq N$ and let v_j be the median lumpy tie over $P(v_s, v_r)$. Then $\text{Lumpy}(S, v_j, P(v_s, v_r))$ is an EF1 allocation of the items in $P(v_s, v_r)$ to S . Further, each agent in S weakly prefers their bundle to $L(v_j)$ and $R(v_j)$.*

The algorithm is specified in Figure 2. It alternately moves a left pointer ℓ (in Steps 2 and 3) and a right pointer r (in Step 4). See [5] for a proof of the following theorem.

► **Theorem 12.** *The moving-knife protocol in Figure 2 finds an EF1 allocation for three agents and runs in $O(|V|)$ time, when G is traceable.*

5 EF2 existence for any number of agents

For two or three agents, we have seen algorithms that are guaranteed to find an EF1 allocation on a path (and on traceable graphs). Both algorithms were adaptations of procedures that identify envy-free divisions in the cake-cutting problem. For the case of four or more agents, we face a problem: there are no known procedures that find connected envy-free division in cake-cutting if the number of agents is larger than three. However, in the divisible setting, a non-constructive existence result is known: Su [29] proved, using Sperner's lemma, that for any number of agents, a connected envy-free division of a cake always exists. One might try to use this result as a black box to obtain a fair allocation for the indivisible problem on a path: Translate an indivisible instance with additive valuations into a divisible cake (where each item corresponds to a region of the cake), obtain an envy-free division of the cake, and round it to get an allocation of the items. Suksompong [30] followed this approach and showed that the result is an allocation where any agent i 's envy $u_i(A(j)) - u_i(A(i))$ is at most $2u_{\max}$, where u_{\max} is the maximum valuation for a single item.

In this section, rather than using Su's [29] result as a black box, we directly apply Sperner's lemma to the indivisible problem. This allows us to obtain a stronger fairness guarantee: We show that on paths (and on traceable graphs), there always exists an EF2

Discrete moving-knife protocol for $n = 3$ agents on a sequence $P = (v_1, v_2, \dots, v_m)$:
 An agent $i \in N$ is a *shouter* if $u_i(L) \geq u_i(M)$ and $u_i(L) \geq u_i(R)$.

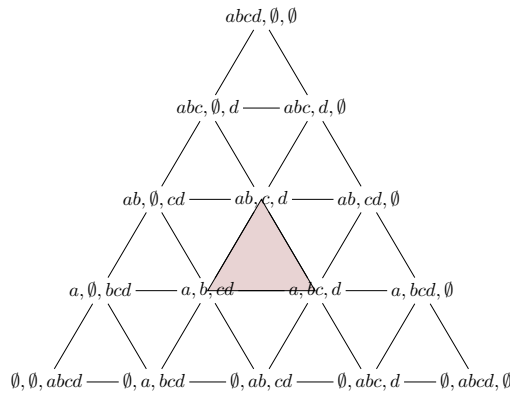
- Step 1.* Initialize $\ell = 0$ and set r so that v_r is the median lumpy tie over the subsequence $P(v_2, v_m)$. Initialize $L = \emptyset$, $M = \{v_2, v_3, \dots, v_{r-1}\}$, and $R = \{v_{r+1}, v_{r+2}, \dots, v_m\}$.
- Step 2.* Add an additional item to L , i.e., set $\ell = \ell + 1$ and $L = \{v_1, v_2, \dots, v_\ell\}$. If no agent shouts, go to Step 3. If some agent s_{left} shouts, s_{left} receives the left bundle L . Allocate the remaining items according to $\text{Lumpy}(N \setminus \{s_{\text{left}}\}, v_r, P(v_{\ell+1}, v_m))$.
- Step 3.* Delete the left-most point of the middle bundle, i.e., set $M = \{v_{\ell+2}, v_{\ell+3}, \dots, v_{r-1}\}$. If the number of shouters is smaller than two, go to Step 4. If at least two agents shout, we show (next page) that there is a shouter s who is a middle agent over $P(v_{\ell+1}, v_m)$. Then, allocate L to a shouter s_{left} distinct from s . Let the agent c distinct from s and s_{left} choose his preferred bundle among $\{v_{\ell+1}\} \cup M$ and $\{v_r\} \cup R$. Agent s receives the other bundle.
- Step 4.* If v_r is the median lumpy tie over $P(v_{\ell+2}, v_m)$, directly move to the following cases (a)–(d). If v_r is not the median lumpy tie over $P(v_{\ell+2}, v_m)$, set $r = r + 1$, $M = \{v_{\ell+2}, v_{\ell+3}, \dots, v_{r-1}\}$, and $R = \{v_{r+1}, v_{r+2}, \dots, v_m\}$; then, go to cases (a)–(d).
- If at least two agents shout, find a shouter s who did not shout at the previous step. If there is a shouter s_{left} who shouted at the previous step, s_{left} receives L ; else, give L to an arbitrary shouter s_{left} distinct from s . The agent c distinct from s and s_{left} choose his preferred bundle among $\{v_{\ell+1}\} \cup M$ and $\{v_r\} \cup R$, breaking ties in favor of the former option. Agent s receives the other bundle.
 - If v_r is the median lumpy tie over $P(v_{\ell+2}, v_m)$ and only one agent s_{left} shouts, give $L \cup \{v_{\ell+1}\}$ to s_{left} and allocate the rest according to $\text{Lumpy}(N \setminus \{s_{\text{left}}\}, v_r, P(v_{\ell+2}, v_m))$.
 - If v_r is the median lumpy tie over $P(v_{\ell+2}, v_m)$ but no agent shouts, go to Step 2.
 - Otherwise v_r is not the median lumpy tie over $P(v_{\ell+2}, v_m)$: Repeat Step 4.

■ **Figure 2** Algorithm producing EF1 allocations for three agents.

allocation.⁴ An allocation is EF2 if any agent's envy can be avoided by removing up to two items from the envied bundle. Again, we only allow removal of items if this operation leaves a connected bundle. For example, on a path, if agent i envies the bundle of agent j , then i does not envy that bundle once we remove its two endpoints.

► **Definition 13** (EF2: envy-freeness up to two outer goods). An allocation A satisfies EF2 if for any pair $i, j \in N$ of agents, either $|A(j)| \leq 1$, or there are two goods $u, v \in A(j)$ such that $A(j) \setminus \{u, v\}$ is connected and $u_i(A(i)) \geq u_i(A(j) \setminus \{u, v\})$.

⁴ To see that EF2 is a stronger property than bounding envy up to $2u_{\max}$, consider a path of four items and two agents with additive valuations 1–10–2–2. The allocation (1, 10–2–2) is not EF2, but the first agent has an envy of $13 < 20 = 2u_{\max}$.



Let us first give a high-level illustration with three agents of how Sperner’s lemma can be used to find low-envy allocations. Given a path, say $P = (a, b, c, d)$, the family of connected partitions of P can naturally be arranged as the vertices of a subdivided simplex, as in the figure on the right. For each of these partitions, each agent i labels the corresponding vertex by the index of a bundle from that partition that i most-prefers. For example, the top vertex will be labelled as “index 1” by all agents, since they all most-prefer the left-most bundle in $(abcd, \emptyset, \emptyset)$. Now, Sperner’s lemma will imply that at least one of the simplices (say the shaded one) is “fully-labeled”, which means that the first agent most-prefers the left-most bundle at one vertex, the second agent most-prefers the middle bundle at another vertex, and the third agent most-prefers the right-most bundle at the last vertex. Notice that the partitions at the corner points of the shaded simplex are all “similar” to each other (they can be obtained from each other by moving only 1 item). Hence, we can “round” the corner-partitions into a common allocation A^* , say by picking one of the corner partitions arbitrarily and then allocating bundles to agents according to the labels. The resulting allocation has the property that any agents’ envy can be eliminated by moving at most one good.⁵

The argument sketched above does not yield an EF1 nor even an EF2 allocation. Intuitively, the problem is that the connected partitions at the corners of the fully-labeled simplex are “too far apart”, so that no matter how we round the corner partitions into a common allocation A^* , some agents’ bundles will have changed too much, and so we cannot prevent envy even up to one or two goods. In the following, we present a solution to this problem, by considering a finer subdivision: we introduce $n - 1$ knives which move in half-steps (rather than full steps), and which might “cover” an item so that it appears in none of the bundles. The result is that the partial partitions in the corners of the fully-labeled simplex are closer together, and can be successfully rounded into an EF2 allocation A^* .

5.1 Sperner’s lemma

We start by formally introducing Sperner’s lemma. Let $\text{conv}(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k)$ denote the convex hull of k vectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$. An n -simplex is an n -dimensional polytope which is the convex hull of its $n + 1$ main vertices. A k -face of the n -simplex is the k -simplex formed by the span of any subset of $k + 1$ main vertices. A triangulation T of a simplex S is a

⁵ One can generalize this argument to show that on paths, there exists an allocation A satisfying a weak form of EF1: for any $i, j \in [n]$, we have $u_i(I_i \cup \{g_i\}) \geq u_i(I_j \setminus \{g_j\})$ for some items g_i, g_j such that $I_i \cup \{g_i\}$ and $I_j \setminus \{g_j\}$ are connected. For additive valuations, this implies that envy is bounded by $u_i(g_i) + u_i(g_j) \leq 2u_{\max}$, which is Suksompong’s [30] result.

collection of sub- n -simplices whose union is S with the property that the intersection of any two of them is either the empty set, or a face common to both. Each of the sub-simplices $S^* \in T$ is called an *elementary* simplex of the triangulation T . We denote by $V(T)$ the set of vertices of the triangulation T , i.e., the union of vertices of the elementary simplices of T .

Let T be some fixed triangulation of an $(n-1)$ -simplex $S = \text{conv}(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n)$. A *labeling function* is a function $L : V(T) \rightarrow [n]$ that assigns a number in $[n]$ (called a *color*) to each vertex of the triangulation T . A labeling function L is called *proper* if

- For each main vertex \mathbf{v}_i of the simplex, L assigns color i to \mathbf{v}_i : $L(\mathbf{v}_i) = i$; and
 - $L(\mathbf{v}) \neq i$ for any vertex $\mathbf{v} \in V(T)$ belonging to the $(n-2)$ -face of S not containing \mathbf{v}_i .
- Sperner's lemma states that if L is a proper labeling function, then there exists an elementary simplex of T whose vertices all have different labels.

We will consider a generalized version of Sperner's lemma, proved, for example, in [2]. In this version, there are n labeling functions L_1, \dots, L_n , and we are looking for an elementary simplex which is *fully-labeled* for some way of assigning labeling functions to vertices, where we must use each labeling function exactly once. The formal definition is as follows.

► **Definition 14** (Fully-labeled simplex). Let T be a triangulation of an $(n-1)$ -simplex, and let L_1, \dots, L_n be labeling functions. An elementary simplex S^* of T is *fully-labeled* if we can write $S^* = \text{conv}(\mathbf{v}_1^*, \mathbf{v}_2^*, \dots, \mathbf{v}_n^*)$ such that there exists a permutation $\phi : [n] \rightarrow [n]$ with

$$L_i(\mathbf{v}_i^*) = \phi(i) \quad \text{for each } i \in [n].$$

► **Lemma 15** (Generalized Sperner's Lemma). Let T be a triangulation of an $(n-1)$ -simplex S , and let L_1, \dots, L_n be proper labeling functions. Then there is a fully-labeled simplex S^* of T .

5.2 Existence of EF2 allocations

Suppose that our graph G is a path $P = (1, 2, \dots, m)$, where the items are named by integers. We assume that $m \geq n$, so that there are at least as many items as agents (when $m < n$ it is easy to find EF1 allocations). Our aim is to cut the path P into n intervals (bundles) $I_*^1, I_*^2, \dots, I_*^n$. Throughout the argument, we use superscripts to denote indices of bundles; index 1 refers to the left-most bundle and index n refers to the right-most bundle.

Construction of the triangulation. Consider the $(n-1)$ -simplex

$$S_m = \{ \mathbf{x} \in \mathbb{R}^{n-1} : \frac{1}{2} \leq x^1 \leq x^2 \leq \dots \leq x^{n-1} \leq m + \frac{1}{2} \}. \quad (5)$$

We construct a triangulation T_{half} of S_m whose vertices $V(T_{\text{half}})$ are the points $\mathbf{x} \in S_m$ such that each x^j is either integral or half-integral, namely,

$$V(T_{\text{half}}) = \{ \mathbf{x} \in S_m : x^j \in \{ \frac{1}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}, \dots, m, m + \frac{1}{2} \} \text{ for all } j \in [n] \}.$$

For reasons that will become clear shortly, we call a vector $\mathbf{x} \in V(T_{\text{half}})$ a *knife position*.

Using Kuhn's triangulation [19, 26, 14], we construct T_{half} so we can write each elementary simplex $S' \in T_{\text{half}}$ as $S' = \text{conv}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ and there is a permutation $\pi : [n] \rightarrow [n]$ with

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \frac{1}{2} \mathbf{e}^{\pi(i)} \quad \text{for each } i \in [n-1], \quad (6)$$

where $\mathbf{e}^j = (0, \dots, 1, \dots, 0)$ is the j -th unit vector. We give an interpretation of (6) shortly.

Each vertex $\mathbf{x} = (x^1, x^2, \dots, x^{n-1}) \in V(T_{\text{half}})$ of the triangulation T_{half} corresponds to a partial partition $A(\mathbf{x}) = (I^1(\mathbf{x}), I^2(\mathbf{x}), \dots, I^n(\mathbf{x}))$ of P where $I^j(\mathbf{x}) := \{y \in \{1, 2, \dots, m\} : x^{j-1} < y < x^j\}$, writing $x^0 = \frac{1}{2}$ and $x^n = m + \frac{1}{2}$ for convenience. Intuitively, \mathbf{x} specifies the

location of $n - 1$ knives that cut P into n pieces. If x^j is integral, that is $x^j \in \{1, \dots, m\}$, then the j -th knife “covers” the item x^j , which is then part of neither $I^j(\mathbf{x})$ nor $I^{j+1}(\mathbf{x})$. This is why $A(\mathbf{x})$ is a *partial* partition. Since there are only $n - 1$ knives but $m \geq n$ items, not all items are covered, so at least one bundle is non-empty.

Property (6) means that, if we visit the knife positions $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ at the corners of an elementary simplex in the listed order, then at each step exactly one of the knives moves by half a step, and each knife moves only at one of the steps.

Construction of the labeling functions. We now construct, for each agent $i \in [n]$, a labeling function $L_i : V(T_{\text{half}}) \rightarrow [n]$. The function L_i takes as input a vertex \mathbf{x} of the triangulation T_{half} (interpreted as the partial partition $A(\mathbf{x})$), and returns a color in $[n]$. The color will specify the index of a bundle in $A(\mathbf{x})$ that agent i likes most. Formally,

$$L_i(\mathbf{x}) \in \{j \in [n] : u_i(I^j(\mathbf{x})) \geq u_i(I^k(\mathbf{x})) \text{ for all } k \in [n]\}.$$

If there are several most-preferred bundles in $A(\mathbf{x})$, ties can be broken arbitrarily. However, we insist that the index $L_i(\mathbf{x})$ always corresponds to a non-empty bundle (this can be ensured since $A(\mathbf{x})$ always contains a non-empty bundle, and u_i is monotonic).

The labeling functions L_i are proper. For each $j \in [m]$, the main vertex \mathbf{v}_j of the simplex S_m has the form $\mathbf{v}_j = (\frac{1}{2}, \dots, \frac{1}{2}, m + \frac{1}{2}, \dots, m + \frac{1}{2})$, where the first $j - 1$ entries are $\frac{1}{2}$ and the rest are $m + \frac{1}{2}$. In the partition $A(\mathbf{v}_j)$, the bundle $I^j(\mathbf{v}_j)$ contains all the items, so is most-preferred (since u_i is monotonic and by our tie-breaking), and so $L_i(\mathbf{v}_j) = j$. Further, any vertex \mathbf{x} belonging to the $(n - 2)$ -face of S_m not containing \mathbf{v}_j satisfies $x^{j-1} = x^j$, and thus in partition $A(\mathbf{x})$, bundle $I^j(\mathbf{x})$ is empty, hence is *not* selected, and so $L_i(\mathbf{x}) \neq j$.

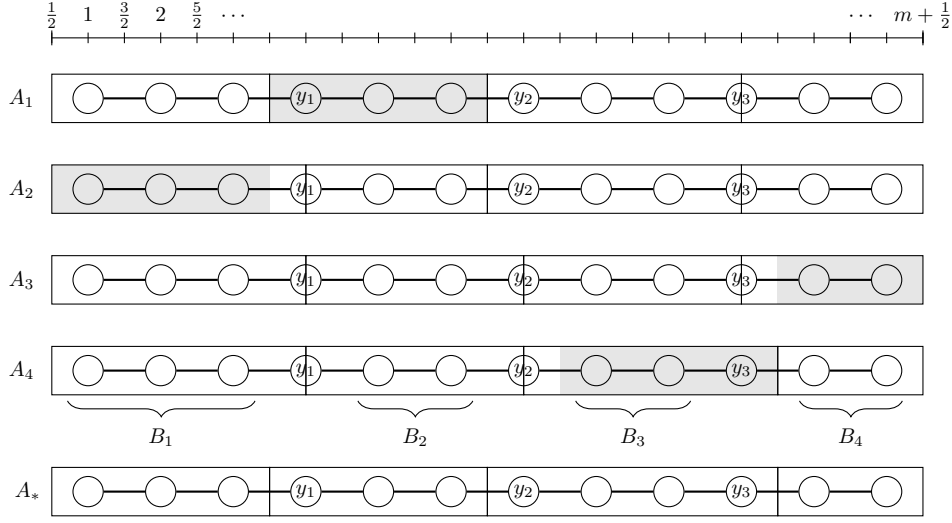
By the generalized version of Sperner’s lemma (Lemma 15), there exists an elementary simplex $S^* = \text{conv}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ of the triangulation T_{half} which is fully-labeled, so that, for some permutation $\phi : [n] \rightarrow [n]$, we have $L_i(\mathbf{x}_i) = \phi(i)$ for all $i \in [n]$.

Translation into partial partitions. The fully-labeled elementary simplex S^* corresponds to a sequence (A_1, A_2, \dots, A_n) of partial partitions of P , which we call the *Sperner sequence*, where $A_i = (I_i^1, \dots, I_i^n) := A(\mathbf{x}_i)$ for each $i \in [n]$. An example of a Sperner sequence is shown in Figure 3. From the labeling, for each agent $i \in [n]$, since $L_i(\mathbf{x}_i) = \phi(i)$, the bundle with index $\phi(i)$ in the partition A_i is a best bundle for i :

$$u_i(I_i^{\phi(i)}) \geq u_i(I_i^j) \quad \text{for each } j \in [n]. \tag{7}$$

Now, for each $j \in [n]$, we define the *basic bundle* $B^j := I_1^j \cap \dots \cap I_n^j$ to be the bundle of items that appear in the j -th bundle of every partition in the Sperner sequence. The set of basic bundles is a partial partition. Let us analyze the items between basic bundles. From (6), each of the $n - 1$ knives moves exactly once, by half a step, while passing through the Sperner sequence (A_1, A_2, \dots, A_n) . Thus, the numbers x_1^j, \dots, x_n^j take on two different values, one of which is integral and the other half-integral. We write y^j for the integral value (so $y^j = x_i^j$ for some $i \in [n]$), and call y^j a *boundary item*. The j -th knife covers the item y^j in some, but not all, of the partial partitions in the Sperner sequence. Now, there are two cases:

- (a) $x_1^j = \dots = x_i^j = y^j - \frac{1}{2}$ and $x_{i+1}^j = \dots = x_n^j = y^j$ for some $i \in [n]$, so that y^j never occurs in the j -th bundle in the Sperner sequence but sometimes occurs in the $j + 1$ st bundle, or



■ **Figure 3** Example of the Sperner sequence A_1, \dots, A_4 for $n = 4$, as well as the derived partition A_* . Vertical lines indicate the positions x_i^1, x_i^2, x_i^3 of the knives, $i = 1, \dots, 4$. Shaded in gray, for $i = 1, \dots, 4$, is the bundle $I_i^{\phi(i)}$ selected by agent i as their favorite bundle in A_i .

(b) $x_1^j = \dots = x_i^j = y^j$ and $x_{i+1}^j = \dots = x_n^j = y^j + \frac{1}{2}$ for some $i \in [n]$, so that y^j sometimes occurs in the j -th bundle in the Sperner sequence but never occurs in the $j + 1$ st bundle. Since y^j is sometimes covered by a knife, it is not part of any basic bundle. Note that

$$B^j \subseteq I_i^j \subseteq \{y^{j-1}\} \cup B^j \cup \{y^j\} \quad \text{for every } i, j \in [n]. \quad (8)$$

Rounding into a complete partition. We now construct a complete partition of the path P into the bundles $(I_*^1, I_*^2, \dots, I_*^n)$ which are defined as follows:

$$I_*^j := I_1^j \cup \dots \cup I_n^j \quad \text{for each } j \in [n].$$

Thus, the bundle I_*^j contains the basic bundle B^j , plus all of the boundary items y^{j-1} or y^j that occur in the j -th bundle at some point of the Sperner sequence. Precisely, for each boundary item y^j , $j \in [n - 1]$, the item y^j is placed in bundle I_*^{j+1} in case (a) above, and it is placed in bundle I_*^j in case (b). Thus, every item is allocated to exactly one bundle.

An EF2 allocation. We first show that the partition $(I_*^1, I_*^2, \dots, I_*^n)$ is such that agents' expectations about the value of the bundles I_*^j are approximately correct (up to two items):

$$u_i(I_*^j) \geq u_i(I_i^j) \geq u_i(B^j) \quad \text{for every agent } i \in [n] \text{ and every } j \in [n]. \quad (9)$$

This follows by monotonicity of u_i , since $I_*^j = I_1^j \cup \dots \cup I_n^j \supseteq I_i^j \supseteq B^j$ by (8).

Now, based on the partition, we define an allocation A_* by $A_*(i) = I_*^{\phi(i)}$ for each agent $i \in [n]$. Then A_* satisfies EF2: For any pair $i, j \in [n]$ of agents, we have

$$u_i(A_*(i)) = u_i(I_*^{\phi(i)}) \stackrel{(9)}{\geq} u_i(I_i^{\phi(i)}) \stackrel{(7)}{\geq} u_i(I_i^{\phi(j)}) \stackrel{(9)}{\geq} u_i(B^{\phi(j)}) \stackrel{(8)}{=} u_i(A_*(j) \setminus \{y^{j-1}, y^j\}).$$

Hence, we have proved the main result of this section:

► **Theorem 16.** *On a path, for any number of agents with monotone valuation functions, a connected EF2 allocation exists.*

6 EF1 existence for four agents

We have seen that Sperner's lemma can be used to show EF2 existence for any number of agents. Why does our proof in the previous section only establish EF2, and not EF1? The reason is that agents' expectations about the contents of a bundle might differ by up to *two* goods from what the bundle will actually contain. In the notation of the previous section, an agent i may be presented with a partial partition I_i where the j -th bundle I_i^j is the basic bundle, i.e., $I_i^j = B^j$. The agent then selects their favorite bundle from I_i , implicitly assuming that the j -th bundle in the rounded partition I_* will also equal B^j , i.e., that $I_*^j = B^j$. However, it may happen that in fact $I_*^j = \{y^{j-1}\} \cup B^j \cup \{y^j\}$, and then i envies the agent who receives bundle j by a margin of two goods.

For four agents, we can adapt our argument to achieve EF1. To do this, we both change the way we round the Sperner sequence into an allocation, and we define new labeling functions that better anticipate how a partial partition will be rounded into the final allocation. In this way, agents' expectations about bundles will only ever be wrong up to one good. In crude terms, agents will expect that each of the two interior bundles will be assigned at least one of the boundary items, and the rounding method ensures that this will indeed happen.

Let $n = 4$. Formally, to define the labeling function, for each agent $i \in [n]$ we construct a *virtual valuation function* $\hat{u}_i(\mathbf{x}, j)$ which assigns a value to each bundle $j \in [n]$ of a partial allocation as specified by a vertex $\mathbf{x} \in V(T_{\text{half}})$. The way these virtual valuations are defined differs based on the index j ; in particular, end bundles ($j = 1, 4$) are treated differently from interior bundles ($j = 2, 3$). The virtual valuations are defined as follows, for each $\mathbf{x} \in V(T_{\text{half}})$ and each $i \in [n]$, where the middle row (11) applies to $j = 2$ and $j = 3$:

$$\hat{u}_i(\mathbf{x}, 1) = \begin{cases} u_i(\{1, \dots, x^1 - 1\}) & \text{if } x^1 \in \mathbb{Z}, \\ u_i(\{1, \dots, x^1 - \frac{3}{2}\}) & \text{if } x^1 \notin \mathbb{Z}. \end{cases} \quad (10)$$

$$\hat{u}_i(\mathbf{x}, j) = \begin{cases} u_i^-(\{x^{j-1}, \dots, x^j\}) & \text{if } x^{j-1} \in \mathbb{Z} \text{ and } x^j \in \mathbb{Z}, \\ u_i(I^j(\mathbf{x})) & \text{otherwise.} \end{cases} \quad (11)$$

$$\hat{u}_i(\mathbf{x}, 4) = \begin{cases} u_i(\{x^3 + 1, \dots, m\}) & \text{if } x^3 \in \mathbb{Z}, \\ u_i(\{x^3 + \frac{3}{2}, \dots, m\}) & \text{if } x^3 \notin \mathbb{Z}. \end{cases} \quad (12)$$

Thus, for an interior bundle $j = 2, 3$, if both the items x^{j-1} and x^j to either side of the bundle are covered by a knife, an agent expects that one of these items (the less-valuable one) will be put into bundle I_*^j of the final rounded allocation (recall the definition of u_i^- in equation (1)). For exterior bundles, $j = 1$ (resp. $j = 4$), if the item x^1 (resp. x^3) is not covered by a knife, the agent does not expect the interior item (next to the knife) to belong to the final bundle I_*^j , even though it belongs to the observed bundle I_i^j . Otherwise, the virtual allocations are equal to $u_i(I^j(\mathbf{x}))$, so the agent expects that $I_*^j = I_i^j$. Later, we show that these expectations are correct up to one item.

Using these virtual valuations, we define labelling functions $\hat{L}_i : V(T_{\text{half}}) \rightarrow [n]$ so that

$$\hat{L}_i(\mathbf{x}) \in \{j \in [n] : \hat{u}_i(\mathbf{x}, j) \geq \hat{u}_i(\mathbf{x}, k) \text{ for all } k \in [n]\}.$$

One can check that these valuation functions are still proper.

Again, by Sperner's lemma, there exists an elementary simplex $S^* = \text{conv}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ of the triangulation T_{half} which is fully-labeled according to our new labeling function: there is a permutation $\phi : [n] \rightarrow [n]$, with $\hat{L}_i(\mathbf{x}_i) = \phi(i)$ for all $i \in [n]$. Again, this elementary simplex induces a Sperner sequence (A_1, \dots, A_n) of partial partitions.

To shorten a case distinction, we assume that $y^2 \in I_1^2 \cup I_2^2 \cup I_3^2 \cup I_4^2$, i.e., that the boundary item y^2 appears in the second but not in the third bundle in the Sperner sequence. This assumption is without loss of generality, since by the left-right symmetry of the definition of virtual valuations, if necessary we can reverse the path P and consider the same elementary simplex with vertices ordered in reverse $(\mathbf{x}_4, \mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1)$; it will still be fully-labeled.

With this assumption made throughout the rest of the argument, we now round the Sperner sequence into a complete partition $(I_*^1, I_*^2, I_*^3, I_*^4)$ of P defined as follows:

$$I_*^1 := I_1^1 \cup \dots \cup I_4^1, \quad I_*^2 := I_1^2 \cup \dots \cup I_4^2, \quad I_*^3 := B^3 \cup \{y^3\}, \quad I_*^4 := B^4.$$

Depending on the placement of the boundary item y^1 , we will either have $I_*^1 = B^1$ or $I_*^1 = B^1 \cup \{y^1\}$; and either $I_*^2 = \{y^1\} \cup B^2 \cup \{y^2\}$ or $I_*^2 = B^2 \cup \{y^2\}$. With these choices, each interior bundle ($j = 2, 3$) receives at least one of the boundary items adjacent to it.

The main part of showing that the partition $(I_*^1, I_*^2, I_*^3, I_*^4)$ can be made into an EF1 allocation is an analogue of (9), which shows that agents' expectations about their bundle are approximately correct. The following analogous proposition is proved by case analysis.

► **Proposition 17.** *For each $i \in [n]$ and each $j \in [n]$, we have $u_i(I_*^j) \geq \hat{u}_i(\mathbf{x}_i, j) \geq u_i^-(I_*^j)$.*

Now again, based on the partition, we can define an allocation A_* by $A_*(i) = I_*^{\phi(i)}$ for each agent $i \in [n]$. Thus, each agent i receives the bundle in the complete partition corresponding to i 's most-preferred index $\phi(i)$. We prove that A_* satisfies EF1: For any pair $i, j \in [n]$ of agents, we have

$$\begin{aligned} u_i(A_*(i)) = u_i(I_*^{\phi(i)}) &\geq \hat{u}_i(\mathbf{x}_i, \phi(i)) && \text{by Proposition 17} \\ &\geq \hat{u}_i(\mathbf{x}_i, \phi(j)) && \text{since } \hat{L}_i(\mathbf{x}_i) = \phi(i) \\ &\geq u_i^-(I_*^{\phi(j)}) = u_i^-(A_*(j)). && \text{by Proposition 17} \end{aligned}$$

Hence, we have proved the main result of this section:

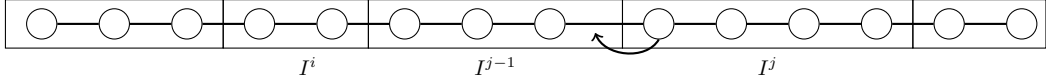
► **Theorem 18.** *On a path, for four agents with monotone valuation functions, a connected EF1 allocation exists.*

For five or more agents, we were not able to construct labeling functions and a rounding scheme which ensure that agents' expectations are correct up to one item. In the four-agent case, each interior bundle is adjacent to an exterior bundle (which helps in the construction), but for five agents, there is a middle bundle whose neighboring bundles are also interior.

7 EF1 existence for identical valuations

A special case of the fair division problem is the case of *identical valuations*, where all agents have the same valuation for the goods: for all agents $i, j \in N$ and every bundle $I \in \mathcal{C}(V)$, we have $u_i(I) = u_j(I)$. We then write $u(I)$ for the common valuation of bundle I . The case of identical valuations often allows for more positive results and an easier analysis. Indeed, we can prove that, for identical valuations and *any* number of agents, an EF1 allocation connected on a path is guaranteed to exist and can be found in polynomial time.

Now, one might guess that in the restricted case of identical valuations, egalitarian allocations are EF1. However, the leximin-optimal connected allocation may fail EF1: Consider a path with five items and additive valuations 1–3–1–1–1 shared by three agents. The unique leximin allocation is (1, 3, 1–1–1), which induces envy even up to one good. The same allocation also uniquely maximizes Nash welfare, so the Nash optimum also does not guarantee EF1. In contrast, when requiring bundles to satisfy matroid constraints (rather than connectivity constraints), the Nash optimum is EF1 with identical valuations [6].



■ **Figure 4** If i envies j even up to one good, Algorithm 1 takes an item out of bundle I^j and moves it in i 's direction.

Algorithm 1 Find a connected EF1 allocation of a path P with identical valuations.

- 1: Let $A = (I^1, \dots, I^n)$ be a leximin allocation of P
 - 2: Fix an agent i with minimum utility in A , i.e., $u(I^i) \leq u(I^j)$ for all $j \in [n]$
 - 3: **for** $j = 1, \dots, i - 1$ **do**
 - 4: **if** i envies I^j even up to one good, i.e., $u(I^i) < u^-(I^j)$ **then**
 - 5: repeatedly delete the right-most item of I^j and add it to I^{j+1} until $u(I^i) \geq u^-(I^j)$
 - 6: **for** $j = n, \dots, i + 1$ **do**
 - 7: **if** i envies I^j even up to one good, i.e., $u(I^i) < u^-(I^j)$ **then**
 - 8: repeatedly delete the left-most item of I^j and add it to I^{j-1} until $u(I^i) \geq u^-(I^j)$
 - 9: **return** A
-

Maximizing an egalitarian objective seemed promising because it ensures that no-one is too badly off, and therefore has not much reason to envy others. The problem is that some bundles might be too desirable. To fix this, we could try to reallocate items so that no bundle is too valuable. This is exactly the strategy of our algorithm: It starts with a leximin allocation, and then moves items from high-value bundles to lower-value bundles, until the result is EF1. In more detail, the algorithm identifies one agent i who is worst-off in the leximin allocation, and then adjusts the allocation so that i does not envy any other bundle up to one good. The algorithm does this by going through all bundles in the allocation, outside-in, and if i envies a bundle I^j even up to one good, it moves one item from I^j inwards (in i 's direction), see Figure 4. As we will show, a key invariant preserved by the algorithm is that the value of I^i never increases, and i remains worst-off. Thus, since i does not envy others up to one good, the allocation at the end is EF1.

Formally, a *leximin allocation* is an allocation which maximizes the lowest utility of an agent; subject to that it maximizes the second-lowest utility, and so on. In particular, if the highest achievable minimum utility is u_L , then the leximin allocation is such that every agent has utility at least u_L , and the number of agents with utility exactly u_L is minimum.

► **Theorem 19.** *For identical valuations on a path, Algorithm 1 finds an EF1 allocation.*

Proof. For an allocation $A = (I^1, \dots, I^n)$, write $u_L(A) := \min_{j \in [n]} u(I^j)$ for the minimum utility obtained in A , and write $L(A) := \{j \in [n] : u(I^j) = u_L(A)\}$ for the set of agents (*losers*) who obtain this utility. For the leximin allocation A_{leximin} obtained at the start of the algorithm, write $u_L^* := u_L(A_{\text{leximin}})$ and $L^* := L(A_{\text{leximin}})$. Note that by leximin-optimality, for every allocation A we must have $u_L(A) \leq u_L^*$, and if $u_L(A) = u_L^*$ then $|L(A)| \geq |L^*|$. Let $i \in L^*$ be the agent fixed at the start of the algorithm.

Claim 1. Throughout the algorithm, $u_L(A) = u_L^*$ and $L(A) = L^*$.

The claim is true before we start the for-loops. Suppose the claim holds up until some iteration of the first for-loop, and we now move an item from I^j to I^{j+1} , obtaining the new bundles I_{new}^j and I_{new}^{j+1} in the new allocation A_{new} . Then $u(I_{\text{new}}^j) \geq u^-(I^j) > u(I^i) = u_L^*$, where the strict inequality holds by the if- and until-clauses. Since no agent other than j has become worse-off in A_{new} , it follows that $u_L(A_{\text{new}}) \geq u_L(A) = u_L^*$. As noted, by optimality of u_L^* , we have $u_L(A_{\text{new}}) \leq u_L^*$. Hence $u_L(A_{\text{new}}) = u_L^*$. Thus, by optimality of

L^* , we have $|L(A_{\text{new}})| \geq |L^*|$. Because agent j has not become a loser (since $u(I_{\text{new}}^j) > u_L^*$ as shown before) and no other agent has become a loser, we have $L(A_{\text{new}}) \subseteq L(A) = L^*$. Thus $L(A_{\text{new}}) = L^*$, as required. The second for-loop is handled similarly.

Claim 2. After both for-loops terminate, agent i does not envy any agent up to one good.

For any $j \neq i$, agent i does not envy j up to one good immediately after the relevant loop has handled j , and at no later stage of the algorithm does I^j change.

It follows that the allocation A returned by the algorithm is EF1: By Claim 1, we have $i \in L(A)$, so that $u(I^j) \geq u(I^i)$ for all $j \in [n]$. By Claim 2, agent i does not envy any other agent up to one good, so that $u(I^i) \geq u^-(I^k)$ for all $k \in [n]$. Hence, for all $j, k \in [n]$, we have $u(I^j) \geq u^-(I^k)$, that is, no agent envies another agent up to one good. ◀

Algorithm 1 can be implemented to run in polynomial time, because with identical valuations, one can use dynamic programming to find a leximin allocation in time $O(m^2n^2)$, and the remainder of Algorithm 1 takes time $O(mn)$, as each item is moved at most n times.

The reallocation stage of our algorithm bears some similarity to Suksompong's [30, Thm. 2] proof that a u_{\max} -equitable allocation exists. In a very recent paper, Oh et al. [23, Lem. C.2] proved independently, using an inductive argument, that EF1 allocations on a path exist for identical valuations, and can be found in polynomial time.

8 Conclusion

We have studied the existence of EF1 allocations under connectivity constraints imposed by an undirected graph. We have shown that for two, three, or four agents, an EF1 allocation exists if the graph is traceable. For any number of agents, we also proved that traceable graphs guarantee the existence of an EF2 allocation, and they guarantee the existence of an EF1 allocation with identical valuations.

Several questions are open. We did not settle whether EF1 allocations on a path exist for five or more agents. Our Sperner technique for four agents seems to not extend to five agents. Extensive sampling did not find an example where no EF1 allocation exists.

Many of our procedures admit efficient implementations for finding fair allocations, but for our Sperner results we do not know better algorithms than searching through all connected allocations. For divisible cake-cutting, it is PPAD-complete to find an envy-free allocation [14]. What is the complexity of finding an EF1 or EF2 allocation on a path?

In the setting without connectivity constraints, it is possible to achieve efficiency and fairness simultaneously: the maximum Nash welfare solution yields an allocation that is both EF1 and Pareto-optimal [13, 4]. In our model, this is unfortunately impossible, since on a path there are instances where there is no connected allocation which is EF1 and Pareto-optimal [18], and it is NP-hard to decide whether such an allocation exists [18].

In this paper, we have only considered *goods*, with monotonic valuations. The setting where some or all items are undesirable (so-called *chores*) is also of interest [7, 22, 27, 1]. On a path, a connected *Prop1* allocation always exists [1], but the existence of EF1 or EF2 allocations in this domain is open. For cake-cutting, when agents consider some parts of the cake undesirable, Sperner's lemma does not directly produce a connected envy-free allocation [27], but other methods can prove the existence of such allocations in most cases [27, 22].

References

- 1 H. Aziz, I. Caragiannis, and A. Igarashi. Fair allocation of combinations of indivisible goods and chores. *CoRR*, 2018. arXiv:1807.10684.
- 2 R. B.apat. A constructive proof of a permutation-based generalization of Sperner's Lemma. *Mathematical Programming*, 44(1):113–120, 1989.

- 3 J. B. Barbanel and S. J. Brams. Cake division with minimal cuts: envy-free procedures for three persons, four persons, and beyond. *Mathematical Social Sciences*, 48(3):251–269, 2004.
- 4 S. Barman, S. K. Krishnamurthy, and R. Vaish. Finding fair and efficient allocations. In *Proceedings of the 19th ACM Conference on Economics and Computation (ACM-EC)*, pages 557–574, 2018.
- 5 V. Bilò, I. Caragiannis, M. Flammini, A. Igarashi, G. Monaco, D. Peters, C. Vinci, and W. S. Zwicker. Almost Envy-Free Allocations with Connected Bundles. *CoRR*, 2018. arXiv:1808.09406.
- 6 A. Biswas and S. Barman. Fair division under cardinality constraints. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 91–97, 2018.
- 7 A. Bogomolnaia, H. Moulin, F. Sandomirskiy, and E. Yanovskaya. Dividing goods and bads under additive utilities. *CoRR*, 2016. arXiv:1610.03745.
- 8 J. A. Bondy and U. S. R. Murty. *Graph Theory*. Springer, 1st edition, 2008.
- 9 S. Bouveret, K. Cechlárová, E. Elkind, A. Igarashi, and D. Peters. Fair division of a graph. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 135–141, 2017.
- 10 S. J. Brams and A. D. Taylor. *Fair division: from cake-cutting to dispute resolution*. Cambridge University Press, 1996.
- 11 S. J. Brams, A. D. Taylor, and W. S. Zwicker. A moving-knife solution to the four-person envy-free cake-division problem. *Proceedings of the American Mathematical Society*, 125(2):547–554, 1997.
- 12 E. Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.
- 13 I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang. The unreasonable fairness of maximum Nash welfare. In *Proceedings of the 17th ACM Conference on Economics and Computation (ACM-EC)*, pages 305–322, 2016.
- 14 X. Deng, Q. Qi, and A. Saberi. Algorithmic solutions for envy-free cake cutting. *Operations Research*, 60(6):1461–1476, 2012.
- 15 L. E. Dubins and E. H. Spanier. How to cut a cake fairly. *The American Mathematical Monthly*, 68(1):1–17, 1961.
- 16 S. Even and R. E. Tarjan. Computing an st-numbering. *Theoretical Computer Science*, 2(3):339–344, 1976.
- 17 J. Hopcroft and R. Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.
- 18 A. Igarashi and D. Peters. Pareto-optimal allocation of indivisible goods with connectivity constraints. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, 2019. arXiv:1811.04872.
- 19 H. W. Kuhn. Some combinatorial lemmas in topology. *IBM Journal of Research and Development*, 4(5):518–524, 1960.
- 20 A. Lempel, S. Even, and I. Cederbaum. An algorithm for planarity testing of graphs. *Theory of Graphs: International Symposium*, pages 215–232, 1967.
- 21 R. J. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (ACM-EC)*, pages 125–131, 2004.
- 22 F. Meunier and S. Zerbib. Envy-free divisions of a partially burnt cake. *CoRR*, 2018. arXiv:1804.00449.
- 23 H. Oh, A. D. Procaccia, and W. Suksompong. Fairly Allocating Many Goods with Few Queries. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, 2019. arXiv:1807.11367.

- 24 A. D. Procaccia. Cake Cutting Algorithms. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 13. Cambridge University Press, 2016.
- 25 J. Robertson and W. Webb. *Cake-cutting algorithms: Be fair if you can*. AK Peters/CRC Press, 1998.
- 26 H. E. Scarf. The computation of equilibrium prices: an exposition. *Handbook of Mathematical Economics*, 2:1007–1061, 1982.
- 27 E. Segal-Halevi. Fairly dividing a cake after some parts were burnt in the oven. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1276–1284, 2018. arXiv:1704.00726.
- 28 W. Stromquist. How to cut a cake fairly. *The American Mathematical Monthly*, 87(8):640–644, 1980.
- 29 F. E. Su. Rental harmony: Sperner’s lemma in fair division. *The American Mathematical Monthly*, 106(10):930–942, 1999.
- 30 W. Suksompong. Fairly allocating contiguous blocks of indivisible items. In *Proceedings of the 10th International Symposium on Algorithmic Game Theory (SAGT)*, pages 333–344, 2017. arXiv:1707.00345.
- 31 R. E. Tarjan. Two streamlined depth-first search algorithms. *CoRR*, 9:85–94, 1986.